



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO EN INFORMÁTICA

Título del proyecto:

“RECONOCIMIENTO DE PATRONES METEOROLÓGICOS
MEDIANTE TÉCNICAS NEURONALES Y ESTADÍSTICAS”

Alumno: Miguel Irigoyen Mancho

Tutor: José Javier Astrain Escola

Pamplona, 9 de Septiembre de 2011

ÍNDICE

1-Introducción	6
1.1-Objetivos	7
1.2-Descripción-resumen del proyecto.....	7
1.3-Formas de abordar la predicción local.....	8
1.4 Herramientas de predicción local existentes	9
2-Fundamentos teóricos.....	12
2.1-Componentes Principales.....	13
2.2-Regresión	16
2.2.1-Regresión lineal	16
2.2.2-Regresión lineal múltiple	18
2.2.3-Regresión no lineal	19
2.3-Clasificadores Neuronales	20
2.3.1-La neurona artificial.....	21
2.3.2-La red neuronal	23
2.3.3-Aprendizaje y validación	24
2.3.4-Perceptron.....	25
2.3.5-Redes multicapa.....	26
2.3.6-Aprendizaje de redes multicapa: Algoritmo de Retropropagación.....	27
2.3.7-Algoritmo de Retropropagación del Error con coeficiente de aprendizaje Adaptativo ...	29
2.3.8-Algoritmo de Levenberg-Marquardt	29
3-Pruebas.....	32
3.1-Variables a predecir	33
3.2-Ficheros de estación.....	35
3.3-Patrones atmosféricos.....	35
3.4-Downscaling.....	43
3.5-Tipos de predicción y validaciones.....	43
3.5.1-predicciones deterministas.....	43
3.5.2-Predicciones probabilísticas	44
3.5.3-Calidad de las predicciones	44
3.5.4-Validación predicciones deterministas	45
3.5.5-Validación Predicciones probabilísticas	46
3.5.6-Predicciones binarias probabilísticas	47

3.5.5-Scripts.....	49
4- resultados Experimentales	56
4.1-Redes neuronales	57
4.1.1-Temperatura	57
4.1.2-Precipitación	73
4.1.3-Vientos	83
4.1.4-Dirección del viento.	92
4.2-Regresión	96
4.2.1-Temperatura	96
4.2.2-Precipitación	102
5.2.3-Viento.....	105
5-Herramienta.....	107
5.1-Observación.....	108
5.2-Históricos	110
5.2.1-Estaciones	110
5.2.2-Patrones	113
5.3-Predicción	115
6-Conclusiones	126
7-Anexo I	128
7.1- Material adjunto.....	128
7.2- Ejecución	128
8-Anexo II.....	129
8.1-Declaración de la topología de la red.....	129
8.2-Entrenamiento.....	129
8.3-Simulación de la red.....	130
9-Referencias y Bibliografía.....	131

1-INTRODUCCIÓN

Conocer qué tiempo va a hacer en los próximos días es algo necesario para el desarrollo normal de nuestras vidas y también de industrias y negocios tan dispares como la agricultura o la obtención de energía, por ello, es muy importante conocer la tendencia meteorológica, por ejemplo, para evitar la pérdida de una cosecha o mejorar el rendimiento de un aerogenerador.

En los últimos años se ha realizado un gran esfuerzo investigador desarrollando mejores y más complejos sistemas de observación que sirven de base para la integración de los modelos numéricos que se utilizan para realizar las predicciones. Muchos de estos modelos numéricos están disponibles para los investigadores por lo que se puede acceder a algunos de ellos, tanto de reanálisis como de predicción.

Debido a la evolución de la tecnología en cuanto a procesadores y memorias no es descabellado poder tratar con este tipo de datos e intentar aplicar técnicas como la estadística, algo que se viene utilizando desde hace bastante tiempo en el mundo de las ciencias atmosféricas, pero parece más interesante la utilización de técnicas más automáticas de aprendizaje pertenecientes a la inteligencia artificial como son las redes neuronales.

1.1- OBJETIVOS

En este proyecto se pretende realizar un estudio del reconocimiento de patrones meteorológicos con el objetivo de poder realizar previsiones meteorológicas para una determinada ciudad a partir de algunas variables meteorológicas de modelos numéricos. Para ello se van a emplear técnicas neuronales y estadísticas de reconocimiento de patrones. Como resultado de este proyecto se construirá una herramienta informática que permita visualizar los datos meteorológicos, realizar las previsiones climatológicas correspondientes y analizar patrones integrando todo lo utilizado a lo largo del desarrollo del proyecto.

1.2-DESCRIPCIÓN-RESUMEN DEL PROYECTO

A la hora de desarrollar el proyecto se realiza una previsión meteorológica de diferentes parámetros (precipitación, temperatura...) para la ciudad de Pamplona a través del reconocimiento de patrones atmosféricos anteriores.

Se definen diferentes patrones, se analizan y comparan los resultados estimados con los datos reales. También se prueban distintas técnicas de previsión, desde los métodos estadísticos más tradicionales como es la regresión lineal hasta la estimación utilizando redes neuronales, para observar cual da mejores resultados.

La estructura del proyecto tiene tres grandes partes. La primera parte trata sobre la detección y caracterización de los diferentes patrones que se van a tener para definir el estado atmosférico. Se utilizarán patrones de modelos numéricos generados por distintas organizaciones e instituciones mediante sondeos, satélites, boyas... Para trabajar con este tipo de datos se obtienen ficheros GRIB [1] que contendrán las distintas variables a considerar.

También se obtienen datos de la estación meteorológica a estudiar o pronosticar. La segunda parte es la del desarrollo del clasificador de patrones meteorológicos. Para ello se emplean técnicas neuronales y en menor medida regresión. Se analizan los resultados obtenidos considerando los distintos patrones obtenidos, valoración de los errores cometidos, estimación de la desviación, etc.

Por último, con todo lo anterior se desarrolla una pequeña aplicación informática con interfaz gráfica para encapsular el clasificador y los datos de entrada para crear una herramienta de predicción sencilla, eficaz y fácil de utilizar. También se integra todo lo utilizado a lo largo de la realización del proyecto.

La implementación de las pruebas como la de la herramienta gráfica se realizará con MATLAB [2].

1.3- FORMAS DE ABORDAR LA PREDICCIÓN LOCAL

Como ya se ha mencionado anteriormente se va resolver el problema de la predicción local aplicando el reconocimiento de patrones meteorológicos utilizando redes neuronales principalmente (también se prueba cómo funciona la regresión lineal múltiple). El método que se utiliza se denomina *downscaling* y se detallará en el apartado 4.4.

Se decide la utilización de redes neuronales debido a que las redes neuronales están especialmente indicadas para extraer conocimiento de conjuntos de datos imprecisos y/o complejos y por su capacidad de aprendizaje. También es un factor a tener en cuenta la originalidad del mismo y a mi parecer puede ser una tendencia de futuro que mejore las técnicas más utilizadas hasta ahora.

Por otro lado también se realizan pruebas con la regresión para comparar un poco los resultados frente a las redes neuronales.

En la actualidad existen diferentes métodos a la hora de realizar la predicción local:

- La **correlación canónica** es una generalización de la regresión lineal multivariada, que tiene en cuenta las relaciones “espaciales” entre las variables que se desean predecir. El planteamiento general del problema es el siguiente: Dada una muestra simultánea de predictores $X = (X_1 \dots X_n)$ y predictandos $Y = (Y_1 \dots Y_m)$, se desea estimar el valor del vector predictando de forma conjunta a partir de los patrones espaciales que mayor correlación presenten con el vector predictor.

Esta técnica trata de suplir las deficiencias de la regresión múltiple cuando se aplica a un conjunto de predictandos correlacionados (por ejemplo, observaciones en un conjunto de estaciones).

- El **método de análogos** introducido por Lorenz (1969) en el marco de la predicción de series temporales es una versión particular de una metodología más general llamada técnica de vecinos próximos (Nearest Neighbors, NN). Esta técnica utiliza el “entorno” del patrón atmosférico previsto para entrenar un modelo local y obtener una predicción. En la práctica, el espacio de patrones atmosféricos se aproxima por la muestra aleatoria de patrones proporcionada por un reanálisis suficientemente amplio.

- Las **redes probabilísticas** combinan grafos y probabilidad para definir modelos que satisfagan un cierto conjunto de relaciones de dependencia. Las relaciones de dependencia entre las variables se expresan mediante un grafo, que es una representación visual e intuitiva; los nodos de grafo representan a las variables y las aristas del grafo las relaciones entre ellas. El grafo proporciona un criterio

para factorizar la Función de Probabilidad Conjunta acorde a las relaciones indicadas en el mismo, incluyendo el mínimo de parámetros necesarios.

Dependiendo del tipo de grafo utilizado (dirigido, no dirigido, mixto) y de la forma de codificar las relaciones de dependencia en el mismo, se tienen distintos tipos de redes probabilísticas. Las más famosas son las **redes Bayesianas** (grafos dirigidos acíclicos) y las **redes de Markov** (grafos no dirigidos).

Para ver con más detalles estos aspectos ver [3], capítulos 3 y 4.

1.4 HERRAMIENTAS DE PREDICCIÓN LOCAL EXISTENTES

Hoy en día es muy normal consultar la predicción local para el municipio en el que se reside o al que se va a acudir por cualquier motivo.

Existen diferentes herramientas automatizadas para poder visualizar la predicción por municipios:

- AEMET (Agencia Estatal de Meteorología).

Predicción por Municipios. Pamplona/Iruña (Navarra)

Predicción

Gráficas

Añadir el tiempo a su web

Capital: Pamplona/Iruña (altitud: 450 m)

Latitud: 42° 48' 50" N - Longitud: 1° 38' 42" O - Posición: Ver localización ▶

Elaborado: miércoles, 31 agosto 2011 a las 09:30 (hora local)

Ver tabla detallada ▶

Descargar XML de la predicción detallada de Pamplona/Iruña

Fecha	mié 31		jue 01				vie 02		sáb 03		dom 04	lun 05	mar 06
	12-18	18-24	0-6	6-12	12-18	18-24	0-12	12-24	0-12	12-24			
Estado del cielo													
Prob. precip.	35%	15%	15%	30%	20%	15%	30%	55%	60%	55%	45%	40%	30%
Cota nieve prov.(m)													
Temp. mín./máx. (°C)	17 / 31		17 / 31				16 / 31		17 / 28		16 / 26	14 / 25	12 / 25
Viento	↑	↙	—	↖	↑	↙	↑	↑	↓	↓	↘	↘	↘
(km/h)	15	10	0	15	5	5	10	10	15	10	15	10	15
Indice UV máximo	7		7				7		7		7		
Avisos	<div></div> <div>Sin Riesgo ▶</div>		<div></div> <div>Sin Riesgo ▶</div>				<div></div> <div>Sin Riesgo ▶</div>						

Ver más municipios de la provincia de Navarra

Ver más municipios de la provincia de Navarra ▶

FIG 1.1 VISUALIZACIÓN AEMET

En la Figura 1.1 se puede ver como se muestra la predicción para una localidad en la web de AEMET [4].

Como se advierte en la propia página:

“Deben interpretarse como la tendencia más probable de la evolución meteorológica.

Se generan de forma automática mediante el tratamiento estadístico de los resultados de modelos numéricos de predicción meteorológica. Esto puede dar lugar ocasionalmente a que no coincidan totalmente con las demás predicciones realizadas por la Agencia, que son elaboradas por predictores que interpretan y adecuan los resultados de los modelos.

La selección se hace por municipios y la información que se presenta son predicciones para la capital del municipio, dado que algunos términos municipales tienen una amplia extensión geográfica, así como diferencias de altura sensibles, algunos de los datos previstos pueden variar de un lugar a otro del mismo.”

Es decir, se generan de manera automática mediante técnicas estadísticas utilizando las salidas del modelo numérico Hirlam, que es el que utiliza AEMET.

Existen otras herramientas del mismo estilo:

Meteored [5]



FIG 1.2 VISUALIZACIÓN METEORED

Eltiempo.es [6]



FIG 1.2 VISUALIZACIÓN ELTIEMPO.ES

También es posible encontrar otras herramientas como Meteosim [7] que asegura que a diferencia de la gran cantidad de predicciones automáticas existentes hoy en día en internet, las cuales utilizan modelos con baja resolución espacial y temporal, Meteosim ofrece la interpretación de las salidas del modelo MASS [8] en alta resolución espacial y temporal, mejorando sustancialmente el producto final.

Como se puede ver existen gran cantidad de herramientas de predicción automatizada pero la mayoría de ellas utilizan técnicas estadísticas.

2-FUNDAMENTOS TEÓRICOS

A continuación se describirán una serie de fundamentos teóricos de los que se hace uso a lo largo del proyecto.

2.1-COMPONENTES PRINCIPALES

Las variables que se utilizan dentro de los patrones pueden estar interrelacionadas y pueden tener dependencia y correlación entre ellas. Puede ser bastante útil en algunas ocasiones utilizar esta dependencia entre los datos para poder representar los datos con un menor número de variables, independientes unas de otras.

El análisis de los componentes principales, también se conoce en meteorología como Funciones Ortogonales Empíricas (EOF), es una técnica estandarizada que permite eliminar la información redundante con una mínima pérdida de variabilidad en los datos. Se logra proyectando el conjunto de datos en un espacio nuevo de dimensión menor que el anterior, en el cual las nuevas variables representan aquellas direcciones del espacio en las que los datos tienen mayor varianza, es decir, es un método para comprimir los datos. Es realmente bueno en espacios de alta dimensionalidad, donde los datos pueden estar correlacionados en sus distintas componentes. Un ejemplo claro de esto son los patrones que se utilizan a distintos niveles de presión (que se utilizan en las pruebas).

Para esto, la base del nuevo espacio se forma con los vectores que cumplen que la muestra proyectada presenta mayor varianza. Estos vectores son de enorme utilidad en meteorología ya que pueden corresponder a patrones dominantes.

Se parte de una muestra de m datos:

$$X_k = (x_{k1} \dots x_{kn})^T, k=1..m$$

En un espacio n dimensional con base canónica $\{e_1, \dots, e_n\}$. Se desea obtener un subespacio de dimensión $d < n$ dado por una nueva base $\{f_1, \dots, f_d\}$ (siendo cada f_j una combinación lineal de los vectores e_i de la base canónica). Para un vector x se tendrá:

$$x = \sum_{i=1}^n e_i x_i \approx \bar{x} = \sum_{i=1}^d f_i \bar{x}_i.$$

El criterio para obtener este subespacio es que la muestra proyectada en el nuevo espacio $\bar{x}_k, k=1..m$ tenga la máxima varianza posible. Es decir, fijada una dimensión d , el problema consiste en encontrar los vectores f_i que proyectan la muestra con varianza máxima. El cálculo matemático para obtener los vectores óptimos es sencillo y consiste en estimar la matriz de varianzas y covarianzas a partir de la muestra de datos. Los autovectores (o Funciones Ortogonales Empíricas, EOFs) de esta matriz son los nuevos vectores f_i y los correspondientes autovalores indican la varianza explicada (la varianza de la muestra proyectada sobre el vector). Los coeficientes de un punto (dato) en la nueva base se denominan Componentes Principales (CPs).

Dada la muestra, se puede estimar la matriz de varianzas y covarianzas C_x , donde cada elemento σ_{ij} representa la covarianza de los datos entre la variable i y la j del espacio original:

$$\sigma_{ij} = \langle (x_{ki} - \mu_i)(x_{kj} - \mu_j) \rangle_k; \mu_i = \langle x_{ki} \rangle_k, \mu_j = \langle x_{kj} \rangle_k,$$

donde $\langle \rangle$ denota la media aritmética. Esta matriz de varianzas y covarianzas es cuadrada y simétrica por lo que se puede calcular una nueva base ortogonal encontrando sus autovalores λ_i (que serán reales) y los correspondientes autovectores f_i :

$$C_x f_i = \lambda_i f_i, i = 1, \dots, n.$$

Es fácil resolver este problema cuando n es pequeño, pero, a medida que aumenta la dimensión, el problema se complica debido al posible mal condicionamiento de la matriz. En estos casos es necesario aplicar métodos numéricos eficientes como la Descomposición en Valores Singulares (SVD), que proporciona una factorización de la matriz C_x de la forma:

$$C_x = F \Lambda F^T,$$

donde Λ es una matriz diagonal que contiene los autovalores λ_i (ordenados de forma decreciente) de C_x , y las columnas de F son los correspondientes autovectores f_i . Además F es una matriz ortogonal y F^T es su inversa. De esta manera, si hacemos la proyección:

$$\bar{x}_k = F^T x_k = \begin{pmatrix} f_{11} & \dots & f_{1n} \\ \vdots & & \vdots \\ f_{n1} & \dots & f_{nn} \end{pmatrix} \begin{pmatrix} x_{1k} \\ \vdots \\ x_{nk} \end{pmatrix}$$

se tendrá el elemento de la muestra proyectado sobre la base de autovectores de C_x , mientras que la proyección inversa se obtendrá mediante $x_k = F \bar{x}_k$. Esta proyección tiene las siguientes propiedades:

- Componentes no correladas $\langle \bar{x}_{ki} \bar{x}_{kj} \rangle_k = 0, i \neq j.$
- $Var(\bar{x}_{ki}) = \lambda_i, i = 1, \dots, n.$
- $\sum_{i=1}^n Var(x_{ki}) = \sum_{i=1}^n Var(\bar{x}_{ki}) = \sum_{i=1}^n \lambda_i.$

Dado que los vectores se eligen en orden decreciente de varianza, es posible hacer un recorte de dimensión reteniendo la máxima varianza posible (obviamente, si se quiere conservar toda la varianza habrá que tomar $d = n$). Si se toman sólo las d primeras EOFs, cada elemento de la muestra se podrá expresar aproximadamente como:

$$\mathbf{x}_k \approx \tilde{\mathbf{F}} \tilde{\mathbf{F}}^T \mathbf{x}_k = \begin{pmatrix} f_{11} & \dots & f_{d1} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ f_{1n} & \dots & f_{dn} \end{pmatrix} \begin{pmatrix} f_{11} & \dots & f_{1n} \\ \vdots & & \vdots \\ f_{d1} & \dots & f_{dn} \end{pmatrix} \mathbf{x}_k,$$

donde $\tilde{\mathbf{F}}$ representa a la matriz \mathbf{F} truncada a los d primeros autovectores. El vector $\bar{\mathbf{x}}_k = \tilde{\mathbf{F}}^T \mathbf{x}_k$ de dimensión $d \times 1$ contendrá las CPs del patrón \mathbf{x}_k , es decir, las componentes del vector en el nuevo espacio de dimensión d . Para recuperar la dimensión original, el vector de CPs se proyectará mediante $\tilde{\mathbf{F}} \bar{\mathbf{x}}_k$, obteniendo una aproximación del vector original (mejor cuanto mayor sea la dimensión d del espacio proyectado).

Maximizar la varianza es equivalente a minimizar la norma cuadrática de los residuos $\bar{\mathbf{x}}_k - \mathbf{x}_k$

$$Var(\mathbf{x}_k) = Var(\mathbf{x}_k + \tilde{\mathbf{F}} \bar{\mathbf{x}}_k - \tilde{\mathbf{F}} \bar{\mathbf{x}}_k) = Var(\mathbf{x}_k - \tilde{\mathbf{F}} \bar{\mathbf{x}}_k) + Var(\tilde{\mathbf{F}} \bar{\mathbf{x}}_k)$$

Por tanto, la técnica de componentes principales obtiene la proyección lineal óptima en sentido de máxima varianza explicada y de mínimo error cuadrático de reconstrucción.

Para eliminar el efecto de las distintas escalas de cada una de las componentes del vector, es conveniente estandarizar los datos (componente a componente) como paso previo a realizar el análisis. De esta forma se evita que las variables de mayor varianza se hagan dominantes en el análisis.

2.2-REGRESIÓN

La regresión es una técnica estadística muy utilizada en la práctica ya que permite inferir relaciones funcionales entre diferentes variables partiendo de una muestra de ellas y realizar predicciones basándose en ellas.

El planteamiento general podría ser:

Dada una muestra simultánea de variables $X_1..X_p$ (predictores) e Y (predictando), se desea obtener una relación funcional que permita estimar el valor del predictando a partir de los valores de los predictores.

Para realizar esta tarea se pueden utilizar modelos de la forma:

$$\hat{y} = f(x_1 \dots x_n) = f(x)$$

Y obtener f a partir de una muestra $\{(x_k, y_k), k = 1..n\}$, de forma que los residuos $\epsilon_k = y_k - \hat{y}_k$, con $\hat{y}_k = f(x)$ sean mínimos.

2.2.1-REGRESIÓN LINEAL

La regresión línea es el modelo más simple de regresión, y utiliza una relación funcional lineal entre las variables de la siguiente manera:

$$y = a_0 + a_1x_1 + \dots + a_px_p + \epsilon$$

Donde $a_0..a_p$ son los parámetros del modelo y ϵ es una variable aleatoria gaussiana. Algunas veces se consideran variables transformadas para tener en cuenta fluctuaciones periódicas.

En la práctica lo que interesa es estimar los parámetros del modelo a partir de una muestra $\{(x_k, y_k), k = 1, \dots, n\}$. En el caso sencillo de una regresión lineal de una dimensión $y_k = ax_k + b$, los parámetros a y b se pueden estimar utilizando el método de mínimos cuadrados que consiste en minimizar la suma de los cuadrados de los errores:

$$E = \sum_{k=1}^n (\epsilon_k)^2 = \sum_{k=1}^n (y_k - \hat{y}_k)^2 = \sum_{k=1}^n (y_k - b - ax_k)^2$$

Y se obtienen las ecuaciones normales:

$$\frac{\partial M}{\partial b} = 0 \Rightarrow \sum_{k=1}^n y_k = nb + a \sum_{k=1}^n x_k,$$

$$\frac{\partial M}{\partial a} = 0 \Rightarrow \sum_{k=1}^n x_k y_k = b \sum_{k=1}^n x_k + a \sum_{k=1}^n x_k^2,$$

Y a partir de estas se obtiene la solución final:

$$b = \frac{\sum_{k=1}^n y_k}{n} - a \frac{\sum_{k=1}^n x_k}{n} = \bar{y} - a \bar{x}$$

$$a = \frac{n \sum_{k=1}^n x_k y_k - \sum_{k=1}^n x_k \sum_{k=1}^n y_k}{n \sum_{k=1}^n x_k^2 - (\sum_{k=1}^n x_k)^2} = \frac{\sigma_{xy}}{\sigma_x^2}$$

Y por tanto la recta de regresión lineal:

$$\hat{y} = \frac{\sigma_{xy}}{\sigma_x^2} (x - \bar{x}) + \bar{y}$$

Una vez se tiene el modelo, es importante entender las hipótesis implicadas, y comprobar que los datos a analizar no las contradicen. El modelo de regresión lineal implica la relación lineal entre el valor y_k del predictando y los valores $x_{1k} \dots x_{pk}$ de los predictores.

Por otra parte, las observaciones $x_{1k} \dots x_{pk}$ no contienen errores, mientras que las observaciones de y_k se miden con un error aleatorio de media cero y varianza σ^2 constante y no correlado.

En la práctica estas condiciones no se cumplen de manera exacta y hay que controlar la calidad del ajuste entre los puntos observados y_k y la recta de regresión. Pearson introdujo un coeficiente que valora esto:

$$R^2 = \frac{\sum_{k=1}^n (\hat{y}_k - \bar{y})^2}{\sum_{k=1}^n (y_k - \bar{y})^2}.$$

Si $R^2 = 1$, entonces los valores observados y_k y los valores \hat{y}_k estimados coinciden; es decir, todos los valores están sobre la recta de regresión. Teniendo en cuenta la definición de la recta de regresión y la definición de varianza, se tiene que:

$$R^2 = \frac{\frac{\sigma_{xy}^2}{\sigma_x^4} n \sigma_x^2}{n \sigma_y^2} = \frac{\sigma_{xy}^2}{\sigma_x^2 \sigma_y^2}.$$

Esto permite definir la ecuación de regresión en función de R

$$\hat{y} - \bar{y} = \frac{\sigma_{xy}}{\sigma_x^2} (x - \bar{x}) = \frac{R \sigma_y}{\sigma_x} (x - \bar{x}).$$

2.2.2-REGRESIÓN LINEAL MÚLTIPLE

En el caso de la regresión lineal múltiple, es preferible plantear el modelo en forma matricial:

$$y = X\beta + \epsilon,$$

Con

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}; X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{p1} \\ \vdots & & & \\ 1 & x_{1n} & \cdots & x_{pn} \end{pmatrix}; \beta = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_p \end{pmatrix}; \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix},$$

Donde X es la matriz de diseño. De forma similar al caso de una sola dimensión, el método de mínimos cuadrados proporciona una estimación de los parámetros del modelo.

$$\begin{aligned} \epsilon^T \epsilon &= (Y - X\beta)^T (Y - X\beta) \\ &= Y^T Y - \beta^T X^T Y - Y^T X \beta + \beta^T X^T X \beta \\ &= Y^T Y - 2\beta^T X^T Y + \beta^T X^T X \beta. \end{aligned}$$

Para obtener el valor mínimo, se deriva con respecto a β y se iguala la expresión resultante a cero, obteniendo las ecuaciones normales:

$$(X^T X) \hat{\beta} = X^T Y,$$

Donde $\hat{\beta}$ es el estimador de mínimos cuadrados de β . Suponiendo que X es de rango total, la formula anterior, conduce al estimador

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y},$$

que demuestra que $\hat{\beta}$ es una función lineal de \mathbf{Y} .

La matriz de covarianzas $\hat{\beta}$ puede obtenerse mediante el siguiente resultado bien conocido: La matriz de covarianzas Σ_V de un conjunto de variables V que son combinaciones lineales de otro conjunto U (es decir, $V = CU$) es

$$\Sigma_V = C \Sigma_U C^T.$$

Entonces, como $\hat{\beta}$ es una combinación lineal de \mathbf{Y} , se tiene

$$\text{Var}(\hat{\beta}) = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \text{Var}(\mathbf{Y}) \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} = \sigma_y^2 (\mathbf{X}^T \mathbf{X})^{-1},$$

Dado que $\text{Var}(\mathbf{Y}) = \sigma_y^2 \mathbf{I}$ Tenemos:

$$r^2 = \frac{\beta^T \mathbf{X}^T \mathbf{Y} - n \bar{Y}^2}{\mathbf{Y}^T \mathbf{Y} - n \bar{Y}^2}$$

2.2.3-REGRESIÓN NO LINEAL

Existe una gran cantidad de problemas donde nos se cumplen las condiciones de los modelos lineales y por tanto no se pueden aplicar; es necesario considerar funciones más complejas que permitan modelar estas relaciones.

Desgraciadamente, este proceso conduce a sistemas de ecuaciones no lineales, y es necesario utilizar métodos costosos de optimización (descenso de gradiente, etc.) para su resolución. Además, dada la existencia de mínimos locales en la función de error, estos métodos no garantizan la obtención del óptimo global.

Sin embargo, aparte de modelos no lineales específicos para aplicaciones concretas, como el modelo de regresión logística, en la práctica no suele ser posible prefijar una cierta relación funcional para las variables involucradas en un problema.

Por ello, el mayor esfuerzo investigador en este ámbito se ha centrado en desarrollar técnicas de estimación no paramétrica, que tienen una forma funcional flexible en base a un conjunto de parámetros transparentes para el usuario que permiten ajustar el modelo a una clase amplia de funciones (funciones continuas, etc.). Las redes neuronales multicapa son uno de estos modelos de regresión lineal no paramétrica

2.3-CLASIFICADORES NEURONALES

Los clasificadores neuronales son sistemas de clasificación basados en Redes Neuronales (RN). Una de posibles definiciones de RN puede ser:

“Las redes neuronales artificiales son dispositivos entrenables mediante ejemplos, que poseen la capacidad de generalizar”.

Según esta definición, es evidente que este tipo de dispositivos pueden ser utilizados para aprender a realizar una clasificación a partir de un conjunto de ejemplos, es decir, de patrones cuya clasificación es conocida.

Las redes neuronales artificiales constituyen un intento de emulación del comportamiento de los cerebros biológicos. La idea surge al observar la gran aptitud de los cerebros biológicos, y en particular del humano, para manejar información imprecisa o incompleta y realizar tareas que resultan difíciles para los algoritmos de computación convencionales.

El cerebro distribuye el procesamiento de la información en los miles de millones de neuronas de los que consta, manejando automáticamente el increíble flujo de datos que se produce continuamente entre ellas. Cómo lleva a cabo esta tarea es aún un misterio.

Sin embargo, se han descubierto una serie de mecanismos, cada uno de los cuales resuelve alguno de los problemas relacionados con el procesado simultáneo. Los más importantes son:

- La auto-organización.
- La asociación.
- La generalización.

La auto-organización, se refiere a que la red sea capaz de organizarse para aprender a realizar determinadas tareas, quizás mediante una estrategia de colaboración y/o competición entre neuronas, especialización por zonas...

La asociación se refiere a que una red puede aprender a asociar, por ejemplo, las abscisas 1, 2, 3... con las ordenadas 1, 4, 9. Por generalización se entiende que si se enseña a la Red a asociar parejas de datos que siguen una cierta ley y le introducimos como entrada un dato que no corresponda a ninguno de los enseñados, su respuesta será una aproximación razonable (precisa si la red ha sido suficientemente entrenada) a la respuesta que se derivaría de dicha ley. En el ejemplo anterior la ley es el cuadrado, y si le damos como entrada la abscisa 2.2 su respuesta, si está bien entrenada, será próxima a $4.84 = 2.2^2$.

Esta capacidad de generalización se convierte en una de sus mayores utilidades, ya que le confiere un comportamiento muy similar al de los cerebros biológicos. No es preciso que veamos todos los libros existentes para poder reconocer un ejemplar, y sabemos que tanto un águila como una paloma, a pesar de sus diferencias, son aves. Si nos presentan un ser vivo, lo podremos clasificar, aún sin haberlo visto nunca, en algún grupo de los Reinos Animal o Vegetal. Existen por supuesto excepciones que sólo podría reconocer una persona especialmente adiestrada, lo que demuestra que, en esos casos, nuestro aprendizaje ha sido insuficiente (no todos somos zoólogos).

Por último mencionaremos que parece que, en parte, la clave de la inmensa capacidad de procesamiento del cerebro está en el procesado paralelo masivo. De esta manera se explica que siendo su "velocidad de proceso" mucho menor que la de los ordenadores, el cerebro sea capaz de realizar tareas para las que los ordenadores actuales están poco dotados. El número de neuronas del cerebro humano es aproximadamente $15 \cdot 10^9$ y cada una está conectada con otras 10^4 neuronas. Sin embargo, notemos que, con la definición dada al principio de esta sección, entran dentro de los dispositivos denominados redes neuronales estructuras muy sencillas (ej: una sola neurona) con tal de que tengan un mecanismo de aprendizaje.

2.3.1-LA NEURONA ARTIFICIAL

En el modelo básico de neurona artificial, la respuesta, salida o nivel de activación y de la neurona se obtiene así:

$$y = \sigma(\bar{y})$$

$$\bar{y} = \omega_0 + \sum_{i=1}^n \omega_i x_i$$

donde $\sigma()$ es la función de salida o función de activación de la neurona e y es la entrada neta, cuya expresión más sencilla corresponde a la neurona lineal en las entradas, y es la que aparece arriba. La constante ω_0 se denomina bias o término de tendencia, ω_i son los pesos y x_i son las señales que excitan a la neurona. Con esto, la neurona artificial más sencilla y utilizada se puede representar así:

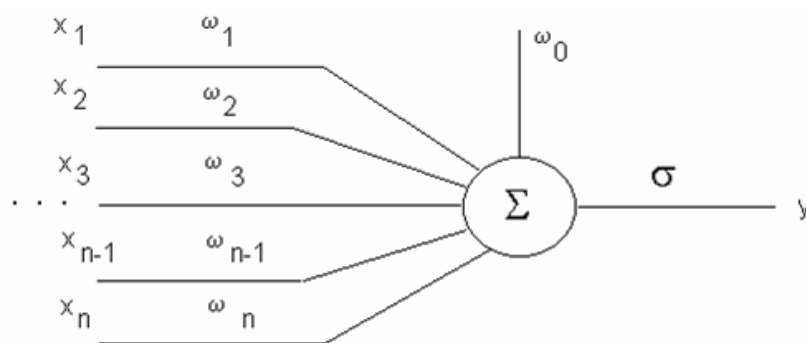


FIG 2.1 NEURONA ARTIFICIAL

Según sea la expresión de la función $\sigma(z)$, obtendremos distintos tipos de neuronas:

- Neurona lineal. $\sigma(z) = z$

Se suele utilizar en la última capa de las redes multicapa, debido al aumento de rango que permite al sistema, aunque puede dar lugar a inestabilidades por no estar acotada su salida.

- Neurona sigmoide: $\sigma(z) = \frac{1}{1+e^{-z}}$

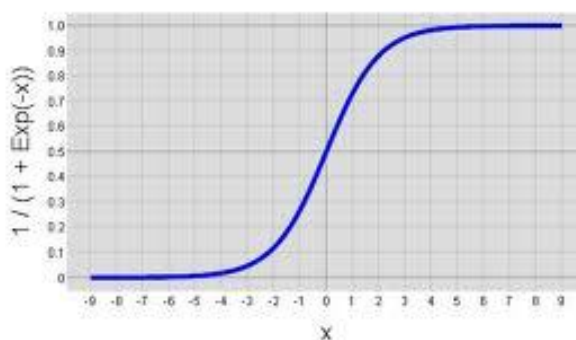


FIG 2.2 FUNCIÓN SIGMOIDE

Esta función, que se representa en la Figura 2.2, es estrictamente creciente (derivada siempre positiva) pero acotada, con valores de salida entre 0 y 1. Es lo que se denomina una función semilineal, que es aproximadamente lineal en valores pequeños de z y no lineal de tipo saturación para valores de z grandes. Las neuronas biológicas presentan este tipo de comportamiento. La función de salida sigmoide es muy utilizada, así como también lo es la función análoga con rango de salida entre -1 y 1 , que es la tangente hiperbólica.

- Neurona tangente hiperbólica: $\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$
- Neurona umbral: la salida es 1 si la entrada neta supera un cierto umbral y 0 (salida binaria) o -1 (salida bipolar) en caso contrario.

2.3.2-LA RED NEURONAL

Red Neuronal es un conjunto de neuronas interconectadas. Las Redes Neuronales artificiales suelen estructurarse en capas. Si definimos el grupo excitador de una neurona como aquellas neuronas que le envían señales, y el grupo receptor como aquellas neuronas que son excitadas por ella, podríamos decir que una capa es un conjunto de neuronas que tienen los mismos grupos excitador y receptor.

Se puede hacer una primera clasificación de las Redes Neuronales Artificiales según el número de capas. Hay redes con una sola capa, que se denominan monocapa, y redes con más de una capa, que se denominan multicapa.

Otra cuestión importante es el sentido de flujo de la información. Hay redes en las que la información fluye en una sola dirección: las neuronas de cada capa no están conectadas entre sí ni con las de capas anteriores, sino sólo con las de capas posteriores.

Éstas se llaman redes de flujo directo o feed-forward. Las redes que no son de flujo directo se llaman realimentadas, recurrentes o feed-back. En ellas, la información puede fluir también hacia atrás, o entre neuronas de la misma capa.

Por último, existen distintos métodos de aprendizaje o ajuste de las conexiones entre neuronas (matriz de pesos de la red), que pueden ser de distintas clases:

- Cálculo analítico previo \Rightarrow red "de pesos programables".
- Aprendizaje por parte de la red \Rightarrow red "de pesos que se aprenden". Dentro de este tipo se pueden distinguir:
 1. Aprendizaje supervisado: la red va ajustando automáticamente sus pesos al serle presentados una serie de ejemplos o asociaciones de (entradas \leftrightarrow salidas) conocidas.
 2. Aprendizaje no supervisado: a la red no se le enseñan explícitamente asociaciones de (entradas \leftrightarrow salidas).

Dentro del aprendizaje no supervisado, se distinguen los métodos de aprendizaje autoorganizativo, que incorporan algún tipo de competición colaboración entre neuronas.

2.3.3-APRENDIZAJE Y VALIDACIÓN

Una de las principales propiedades de las redes neuronales es su capacidad de aprender a partir de datos. Una vez que ha sido elegida la arquitectura de red para un problema particular, los pesos de las conexiones se ajustan para codificar la información contenida en un conjunto de datos de entrenamiento.

Las redes multicapa y las redes recurrentes son apropiadas para problemas de aprendizaje supervisado, donde se dispone de un conjunto de patrones de entrenamiento de la forma $(x_p, y_p) = (x_{1p} \dots x_{mp}; y_{1p} \dots y_{np})$, $p = 1..a$, donde cada patrón de entrada x_p tiene asociado el correspondiente patrón de salida y_p . El entrenamiento está basado en que la red sea capaz de reproducir estos patrones con el menor error posible. Este esquema responde a problemas de clasificación y aproximación o ajuste de datos, y desde un punto de vista estadístico se puede considerar como una generalización de la regresión no lineal, donde el modelo es la propia actividad de la red completa y los parámetros a ajustar son los pesos de las conexiones. Como se verá más adelante, el problema de aprender este tipo de redes se reduce a obtener los pesos apropiados utilizando algún algoritmo de aprendizaje apropiado.

Por otra parte, las redes competitivas responden a problemas de aprendizaje no supervisado, donde se dispone de un conjunto de patrones de entrenamiento de la forma $x_p = (x_{1p} \dots x_{mp})$, $p = 1..a$. En este caso, no se conoce el valor de salida de cada patrón de entrada x_p y la red tiene que auto-organizarse para hallar las salidas óptimas para cada entrada (en base a algún criterio de aprendizaje que corresponda a maximizar la entropía de la salida, etc.).

Normalmente este tipo de redes se utilizan en problemas de agrupamiento, donde cada una de las neuronas de salida representa un grupo distinto, y la red tiene que distribuir los patrones de entrenamiento entre los grupos de forma automática.

Una vez terminado el proceso de aprendizaje y calculados los pesos de la red neuronal, es importante comprobar la calidad del modelo resultante. Por ejemplo, en el caso de aprendizaje supervisado, una medida de la calidad puede darse en términos de los errores entre los valores de salida deseados y los obtenidos por la red neuronal.

Algunas medidas estándar del error son:

1. La suma de los cuadrados de los errores (*Sum of Square Errors, SSE*), definida como:

$$\sum_{p=1}^a ||y_p - \hat{y}_y||^2$$

2. La raíz cuadrada del error cuadrático medio (Root Mean Square Error, RMSE) definida como:

$$\sqrt{\sum_{p=1}^a ||y_p - \hat{y}_y||^2 / n}$$

3. El error máximo:

$$\max_{p=1..a} ||y_p - \hat{y}_y||$$

También es deseable realizar una validación cruzada para obtener una medida de la capacidad de generalización del modelo; es decir, la capacidad de reproducir nuevos patrones del problema, no utilizados en el proceso de aprendizaje.

Con este propósito, los datos disponibles se pueden dividir en dos partes: una parte destinada al entrenamiento de la red y otra parte a la validación.

Cuando el error de validación es significativamente mayor que el error de entrenamiento, entonces se produce un problema de sobreajuste durante el proceso de entrenamiento que puede ser debido a un excesivo número de parámetros. Por tanto, es necesario mantener un compromiso entre la complejidad del modelo, y el número de datos disponible, de forma que el modelo sea más simple que el conjunto de datos que representa. En caso contrario, el modelo sólo memoriza la información no siendo capaz de generalizar lo aprendido a nuevas situaciones.

2.3.4-PERCEPTRON

Los perceptrones (Rosenblat, 1962) son las arquitecturas más simples y consisten en una capa de entrada, $\{x_1 \dots x_m\}$, y una de salida, $\{y_1 \dots y_n\}$, de forma que las neuronas de la capa de salida están conectadas con las de entrada y no hay conexiones entre las neuronas de una misma capa (ver Fig. 2.3). Para indicar el número de entradas y de salidas, este tipo de redes suelen denotarse de forma abreviada mediante $m:n$, siendo m y n el número de neuronas de entrada y salida respectivamente (obsérvese que las neuronas de entrada no realizan ningún cálculo y, por tanto, no contabilizan). Los perceptrones suelen

denominarse también redes de retro-propagación de una única capa (*one layer feedforward network*).

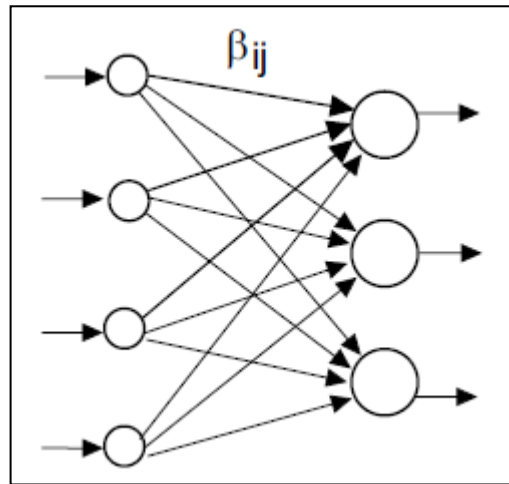


FIG 2.3 PERCEPTRÓN

En un perceptrón, las neuronas de salida típicas, realizan el cálculo:

$$\hat{y}_j = f\left(\sum_{i=0}^m \beta_{ji}\right) = f\left(\beta_j^T x\right), j = 1, \dots, n.$$

Donde $f(\cdot)$ es la función de activación y j el correspondiente vector de pesos p denota el valor de salida obtenido insertando el correspondiente patrón de entrada en la red mientras y_p denota el valor real de salida conocido.

2.3.5-REDES MULTICAPA

El perceptrón es un modelo muy limitado desde el punto de vista práctico y demasiado sencillo desde el punto de vista biológico, como matemático, son modelos equivalentes a la regresión lineal y logística.

Por ello, se han desarrollado modelos más generales (perceptrones multicapa, o redes multicapa) que incorporan capas internas ocultas, además de la capa de entrada y la de salida. Los nodos de la capa de entrada alimentan la red hacia delante donde cada nodo en las capas ocultas y de salida recibe de forma progresiva una entrada de los nodos de las capas previas y calcula un valor de salida para la siguiente capa.

Las capas ocultas dan flexibilidad al modelo resultante, el cual se puede considerar un modelo no paramétrico de regresión no lineal.

Sin embargo, no existe ninguna pauta a la hora de realizar el diseño de estas redes lo que hace difícil realizar la elección sobre el número de capas o el número de neuronas en cada capa.

Un perceptrón con dos capas ocultas es capaz de aproximar una función continua con un grado de exactitud dado, si la función no es continua se necesitan tres capas. En la práctica es necesario probar redes con distintas configuraciones para obtener un modelo apropiado.

Una vez elegido un diseño para la red neuronal, es necesario utilizar un algoritmo de aprendizaje para ajustar los parámetros (pesos) a los datos. El método de aprendizaje más popular para redes multicapa es conocido como retro-propagación (*backpropagation*) y está basado en minimizar la función de error cuadrático total usando el método del descenso de gradiente de forma apropiada, dado que los pesos de las dos capas dependen unos de otros.

2.3.6-APRENDIZAJE DE REDES MULTICAPA: ALGORITMO DE RETROPROPAGACIÓN

El método de *backpropagation* (BP) es el más usado en redes flujo directo con varias capas (capas ocultas), también llamadas redes PERCEPTRON multicapa. Es una generalización del método Delta Rule en la que, en cada ciclo de aprendizaje, el gradiente del error se lleva hacia atrás en las capas dando una regla de variación de los pesos de todas las capas.

Presenta problemas de lentitud y dificultades en el caso de que haya mínimos locales. Para tratar de evitar estos problemas se han desarrollado modificaciones al algoritmo de BP básico.

Algoritmo de Retropropagación del Error ‘estándar’

Se define la variación del peso w_{ij} en el μ -ésimo ciclo de la misma manera que en la Delta Rule, es decir, utilizando el gradiente:

$$w_{ij}^{(m)}(\mu) - w_{ij}^{(m)}(\mu - 1) = -\alpha \sum_{k=1}^N \frac{\partial E_k}{\partial w_{ij}^{(m)}(\mu - 1)}$$

Operando se obtiene:

$$\frac{\partial E_k}{\partial w_{ij}^{(m)}(\mu-1)} = \frac{\partial E_k}{\partial x_{k,i}^{(m)}} \frac{\partial x_{k,i}^{(m)}}{\partial w_{ij}^{(m)}(\mu-1)}$$

Y los factores de la derecha de la igualdad se pueden expresar como:

$$\begin{aligned} \frac{\partial x_{k,i}^{(m)}}{\partial w_{ij}^{(m)}(\mu-1)} &= \frac{df_m(\bar{x}_{k,i}^{(m)})}{d\bar{x}_{k,i}^{(m)}} x_{k,j}^{(m-1)} \\ \frac{\partial E_k}{\partial x_{k,i}^{(m)}} &= \sum_{l=1}^{n_{m+1}} \frac{\partial E_k}{\partial x_{k,l}^{(m+1)}} \frac{df_{m+1}(\bar{x}_{k,l}^{(m+1)})}{d\bar{x}_{k,l}^{(m+1)}} w_{li}^{(m+1)}(\mu-1) \end{aligned}$$

Donde la segunda expresión es la que da nombre al algoritmo, ya que las derivadas del error con respecto a las salidas de una determinada capa se calculan a partir de las derivadas con respecto a salidas de capas posteriores (retropropagación).

El proceso de cálculo comenzará, por tanto, por la última capa, para la cual la derivada del error con respecto a cada salida se obtiene de manera directa. Así, si se utiliza el típico error cuadrático, se tiene:

$$\frac{\partial E_k}{\partial x_{k,i}^{(n)}} = y'_{k,i} - y_{k,i}.$$

En caso de utilizar otro tipo de error se debe obtener la expresión correspondiente. Con este cálculo realizado para todas las neuronas de la última capa se procede a obtener las derivadas del error con respecto a las salidas de la capa anterior, y luego de la anterior, y así sucesivamente, utilizando la expresión:

$$w_{ij}^{(m)}(\mu) - w_{ij}^{(m)}(\mu-1) = -\alpha \sum_{k=1}^N \frac{df_m(\bar{x}_{k,i}^{(m)})}{d\bar{x}_{k,i}^{(m)}} x_{k,j}^{(m-1)} \sum_{l=1}^{n_{m+1}} \frac{\partial E_k}{\partial x_{k,l}^{(m+1)}} \frac{df_{m+1}(\bar{x}_{k,l}^{(m+1)})}{d\bar{x}_{k,l}^{(m+1)}} w_{li}^{(m+1)}(\mu-1)$$

Ya se ha visto que las derivadas totales de la función de activación respecto a su argumento tienen expresiones sencillas.

Al igual que la Regla Delta, el algoritmo de Retropropagación del Error se puede programar realizando un ajuste de pesos tras la presentación de cada ejemplo

entrada/salida. Es decir, suprimiendo en la expresión del gradiente el sumatorio en k . No está demostrado que este algoritmo tienda a un mínimo, ni siquiera que el error descienda después de cada ciclo de presentación de los ejemplos, pero suele hacerlo y, para errores grandes, es más rápido que el algoritmo exacto.

2.3.7-ALGORITMO DE RETROPROPAGACIÓN DEL ERROR CON COEFICIENTE DE APRENDIZAJE ADAPTATIVO

El parámetro α de aprendizaje se varía según la zona de la superficie de error en la que se esté, tratando de acelerar el algoritmo en las zonas en que no haya peligro de desestabilización. El que tiene implementado MATLAB es como sigue:

- $E(\mu) < E(\mu - 1)$, $\alpha_{\mu+1} = 1.05\alpha_{\mu}$.
- $E(\mu - 1) < E(\mu) < 1.04E(\mu - 1)$, $\alpha_{\mu+1} = \alpha_{\mu}$.
- $E(\mu) > 1.04E(\mu - 1)$, $\alpha_{\mu+1} = 0.7\alpha_{\mu}$ y se rectifica la última actualización de los pesos.

Este esquema no sólo acelera el cálculo sino que, al permitir crecer al error, sirve para salvar mínimos locales.

Algoritmo de Retropropagación del Error con “momento”

Este método trata de ayudar al algoritmo a salir de mínimos locales, realizando una búsqueda más global que el algoritmo estándar. Para ello, la modificación de los pesos se hace depender no sólo del valor del gradiente en el punto donde esté sino también de la modificación anterior de los pesos (se suma una fracción del ajuste de peso anterior al ajuste actual).

2.3.8-ALGORITMO DE LEVENBERG-MARQUARDT

Muchas veces es interesante usar algoritmos de aprendizaje de segundo orden, es decir, que utilizan información de la segunda derivada del error con respecto a los pesos. Esta información hace que se tenga en cuenta con más detalle la forma de la superficie de error, acelerando la convergencia al mínimo: se suele alcanzar en menos ciclos de aprendizaje. Por desgracia, el número de cálculos que hay que hacer en cada ciclo aumenta mucho por lo que se deben sopesar, en cada caso, las ventajas e inconvenientes del uso de estos algoritmos.

El algoritmo de minimización de segundo orden básico es el de Newton. No es éste el que se utiliza para el aprendizaje de redes neuronales, sino otros algoritmos basados en él. El más conocido y utilizado es el de Levenberg-Marquadt [9] que se basa en la idea de utilizar un método de Newton aproximado en las cercanías del mínimo y el método del gradiente en puntos alejados del mismo.

El desarrollo comienza aproximando el cálculo del Hessiano [10] (la matriz hessiana de una función f de n variables, es la matriz cuadrada de $n \times n$, de las segundas derivadas parciales) del error. Para el error cuadrático dicho Hessiano se puede expresar así:

$$\mathbf{H}^{ij} = \frac{\partial^2 E}{\partial w^i \partial w^j} = \sum_{l=1}^N \left[\frac{\partial y'_l}{\partial w^i} \frac{\partial y'_l}{\partial w^j} + (y'_l - y_l) \frac{\partial^2 y'_l}{\partial w^i \partial w^j} \right]$$

El segundo sumando de esta expresión se desprecia al considerar que los errores son aleatorios y de-correlacionados con la derivada segunda de la salida, por lo que se puede considerar como un ruido blanco.

A partir del primer sumando del Hessiano se construye la siguiente matriz:

$$\mathbf{J}^{ij} = \sum_{l=1}^N \left[\frac{\partial y'_l}{\partial w^i} \frac{\partial y'_l}{\partial w^j} \right] + \lambda \mathbf{I}_M$$

donde \mathbf{I}_M es la matriz identidad de orden M y λ es un parámetro que hace una función similar al de aprendizaje en el algoritmo de *backpropagation*. La modificación de los pesos con el método de Levenberg-Marquadt en el μ -ésimo ciclo de aprendizaje viene dada por la siguiente expresión:

$$\vec{W}(\mu) = \vec{W}(\mu - 1) - \mathbf{J}^{-1}(\mu - 1) \nabla E(\mu - 1)$$

Notemos que la matriz \mathbf{J} se puede calcular a partir de la siguiente matriz auxiliar:

$$\mathbf{J}\mathbf{a}_l^i = \frac{\partial y'_l}{\partial w^i}$$

Obteniéndose

$$\mathbf{J} = \mathbf{J}\mathbf{a}^T \mathbf{J}\mathbf{a} + \lambda \mathbf{I}_M$$

El cálculo de la matriz auxiliar, como vemos, es el cálculo de la derivada primera de la salida de la red con respecto a los pesos, lo que ya sabemos hacer para todos los pesos de la red mediante retropropagación. Nótese que hasta aquí no son necesarios cálculos más complicados que para el algoritmo de *Backpropagation*.

Es fácil comprobar que la matriz auxiliar $\mathbf{J}\mathbf{a}$ da también el gradiente del error:

$$\nabla E^i = \frac{\partial E}{\partial w^i} = \sum_{l=1}^N \mathbf{J}\mathbf{a}_l^i (y'_l - y_l)$$

3-PRUEBAS

En este capítulo se van a describir todos los elementos necesarios para poder entender cómo se han realizado las pruebas experimentales. Se detalla qué variables se van a predecir, de dónde y cómo se obtienen los datos de la estación y los patrones, los diferentes tipos de predicción que se van a realizar con sus respectivas formas de valorar el error...

3.1-VARIABLES A PREDECIR

A continuación se describen las diferentes variables que se van a intentar pronosticar mediante las redes neuronales y la regresión. Se predicen para Pamplona utilizando los datos disponibles de estaciones (en el siguiente apartado se describe el proceso).

Según el DEPARTAMENTO NAVARRO DE METEOROLOGÍA [18] las variables a predecir se pueden definir así:

Temperatura: La temperatura del aire se podría describir como la mayor o menor agitación de las numerosas moléculas que lo componen. Se mide mediante un sensor de temperatura y humedad, situado habitualmente a 2 metros sobre el nivel del suelo.

Este sensor lleva como accesorio un protector para radiación, que protege al sensor de temperatura de la radiación directa y difusa, ya que, si ésta incidiese directamente sobre él, se registrarían medidas muy superiores a la realidad. Su diseño es tal que permite la ventilación natural.

La temperatura se mide en grados centígrados °C.

En el caso de estas pruebas concretas, se realizará una predicción tanto de temperatura máxima diaria como de la mínima.

Precipitación: Se mide la cantidad de lluvia caída en un intervalo de tiempo. El instrumento de medida es el pluviómetro, y la unidad l/m² (litros por metro cuadrado) o su equivalente mm. (milímetros). La predicción que se llevará a cabo es de l/m² diarios.

Viento: El viento viene caracterizado por dos magnitudes: velocidad y dirección.

La velocidad del viento mide la componente horizontal del desplazamiento del aire en un punto y en un instante determinado. Se mide mediante un anemómetro, y la unidad de medida es habitualmente metros por segundo (m/s). Las ausencias de viento se denominan calmas.

La dirección mide la componente horizontal de la velocidad del viento. En meteorología es importante tener en cuenta que la dirección nos indica de dónde viene el viento, no hacia dónde va. Se mide en grados, desde 0° (excluido) hasta 360° (incluido), girando en el sentido de las agujas del reloj en el plano horizontal visto desde arriba. Valores cercanos a 1° y 360° indican viento del norte, cercanos a 90° viento del este, 180° del sur y 270° del oeste. Entre estos valores tendremos el resto de direcciones: nordeste, sureste, suroeste y noroeste.

Para medir la dirección del viento, además de grados, se utilizan los sectores. Esta magnitud se emplea en el cálculo de los resúmenes diarios, donde se muestra el sector que ha predominado en ese día (moda).

Suponemos una circunferencia dividida en ocho sectores, numerados de 1 a 8 en la dirección de las agujas del reloj. El sector 1 se encuentra cortado en el centro por la

dirección Norte, el 3 por la dirección Este, el 5 por la dirección Sur y el 7 por la dirección Oeste.

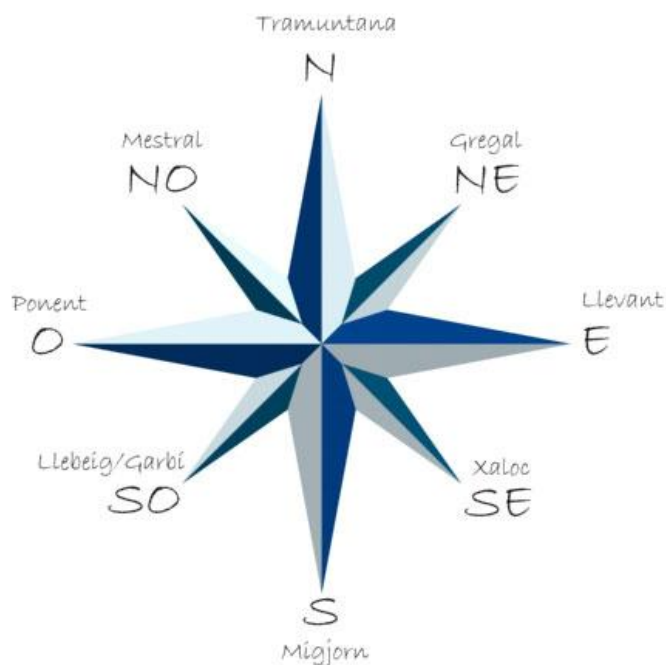


FIG 3.1 ROSA DE VIENTOS

La dirección del viento se mide mediante una veleta.

La velocidad y la dirección del viento se miden a 10 metros de altura sobre el nivel del suelo en la mayoría de las estaciones, y a 2 metros de altura en algunas de ellas.

De esta manera los sectores indican:

- Sector 1: Viento del Norte
- Sector 2: Viento del Nordeste
- Sector 3: Viento del Este
- Sector 4: Viento del Sureste
- Sector 5: Viento del Sur
- Sector 6: Viento del Suroeste
- Sector 7: Viento del Oeste
- Sector 8: Viento del Noroeste

El valor 9 indica que no hay predominio de ninguna dirección.

Para hacerlo más fácil podemos suponer cuatro zonas N-Ne, E-SE, S-SO y O-NO que en las pruebas se han realizado da bastante buen resultado.

3.2-FICHEROS DE ESTACIÓN

Para realizar la predicción se necesitan los datos históricos de estaciones, en este caso se realiza la predicción para la estación de Pamplona. Se descargan los datos disponibles de la estación en ficheros Excel desde meteo navarra [18], se completan algunos datos que faltan desde los página de tu tiempo [19].

Se obtienen datos históricos de precipitación diaria acumulada, temperatura máxima y temperatura mínima desde Agosto de 1957 a Diciembre de 2009. Los ficheros que se generan con los datos son ficheros de Excel y ficheros .dat de donde leerán los scripts en las pruebas, son los siguientes:

- pamplonamax.dat
- pamplonamin.dat
- tmax19892009.dat
- tmin19892009.dat
- lluvia-pamplona.dat
- preci19892009.dat

Del viento no se tienen datos disponibles en la página hasta el año 2000 y los ficheros con datos del viento son:

- pamplonavviento.dat
- vmax.dat
- direccionvientos.dat

En todos los casos se trata de datos diarios, y para el caso de la dirección del viento se toma la moda.

3.3-PATRONES ATMOSFÉRICOS

Los campos resultantes de de las integraciones de los modelos numéricos del tiempo caracterizan el estado de la atmósfera, tanto el estado inicial como la predicciones, y en algunos casos están disponibles para los investigadores.

Por otra parte, también se encuentran disponibles los campos que se producen en proyectos de re análisis en los que se ha integrado un mismo modelo para un periodo de tiempo extenso. En el caso del ECMWF (Centro Europeo de Predicción a Medio Plazo) [25], que es de donde se han obtenido los patrones para las pruebas, tiene tres proyectos, el primer proyecto de reanálisis global llevado a cabo en el ECMWF se denominó ERA-15 y proporciona los campos de análisis y las predicciones a corto plazo cada seis horas obtenidas con un modelo T106L31 (1.125 grados de resolución) para el periodo comprendido entre Diciembre-1978 y Febrero-1994. Este proyecto fue extendido y se

añadió información para el periodo Septiembre 1957 - Agosto 2002 (ERA-40) obtenida con un modelo de mayor resolución (T159L60, aprox. 0.675 grados).

Por último, se ha puesto en marcha el proyecto Era-interim que contiene información desde Enero 1989 hasta pocos meses antes de la fecha actual. Este modelo es el que mayor resolución tiene a la hora de descargar los datos de manera gratuita.

Estos modelos se pueden obtener en ficheros GRIB (*Gridded Binary*) [1] que son archivos que contienen los datos de predicciones. La principal ventaja de estos archivos con respecto a las cartas meteorológicas comunes es que son tremendamente compactos, siendo capaces de incluir mucha información en archivos muy pequeños. Esto los hace especialmente útiles para su utilización como medio de transmisión de información meteorológica en condiciones de difícil comunicación (email vía radio o vía satélite).

Se dispone de información a las 00, 06, 12 y 18 UTC cada día y de variables en diferentes niveles de presión y en superficie. Las variables más utilizadas en altura para definir el estado de la atmósfera son:

- Geopotencial (Z): El geopotencial expresa a qué altura encuentras una presión determinada. El geopotencial se define como $G = g \cdot dz$ donde g es el valor de la gravedad en el punto donde calculamos la altura y dz es la altura. A su vez, se define la altura geopotencial como G/g' donde g' es un valor de referencia fijo de la gravedad terrestre. Si hacemos el cociente sustituyendo G por lo que es en realidad pues queda: $\text{Altura geopotencial} = g \cdot dz / g'$.
- Temperatura (T): (ya definida en las variables).
- Velocidad viento U: componente zonal W-E
- Velocidad viento V: componente meridional N-S
- Humedad relativa (H): La humedad relativa es la relación porcentual entre la cantidad de vapor de agua real que contiene el aire y la que necesitaría contener para saturarse a idéntica temperatura.

Cada una de estas variables se representa los puntos de rejilla de diferente resolución:

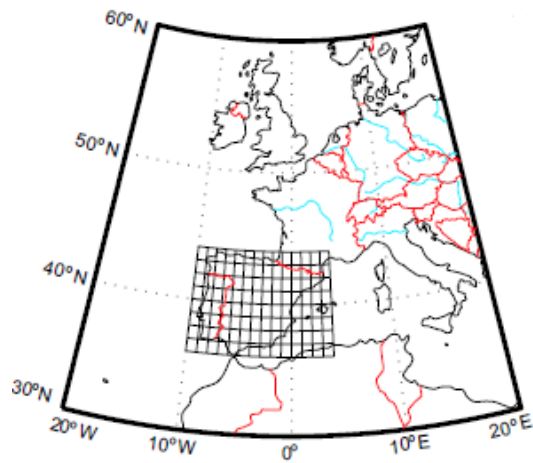


FIG 3.2 EJEMPLO DE REJILLA LATITUD Y LONGITUD

En el caso de este proyecto, se han utilizado los siguientes modelos de rejilla:

Patrón NAO

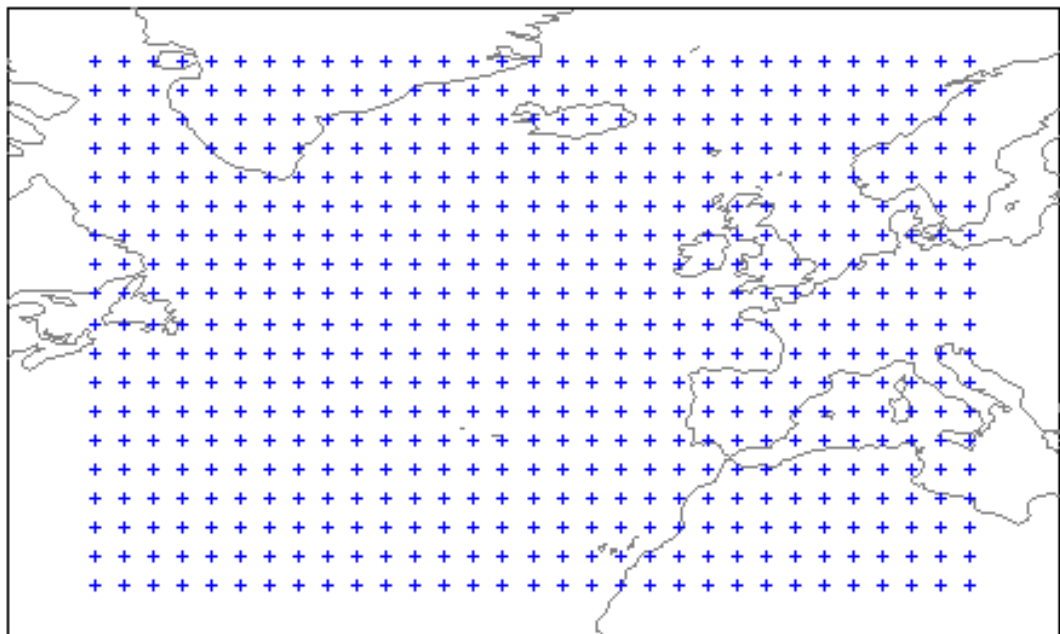


FIG 3.3-PATRÓN NAO 1.5°

Patrón Iberia resolución 1° (solo para variables TZRUV juntas)

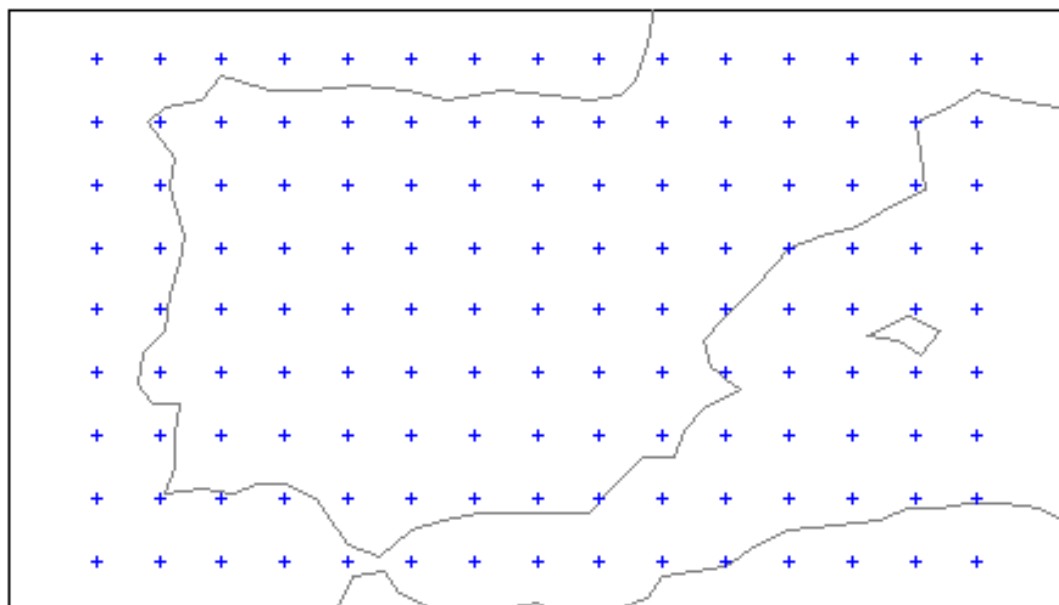


FIG 3.4-IBERIA 1°

Patrón de IBERIA resolución 1,5° (se pueden obtener variables por separado)

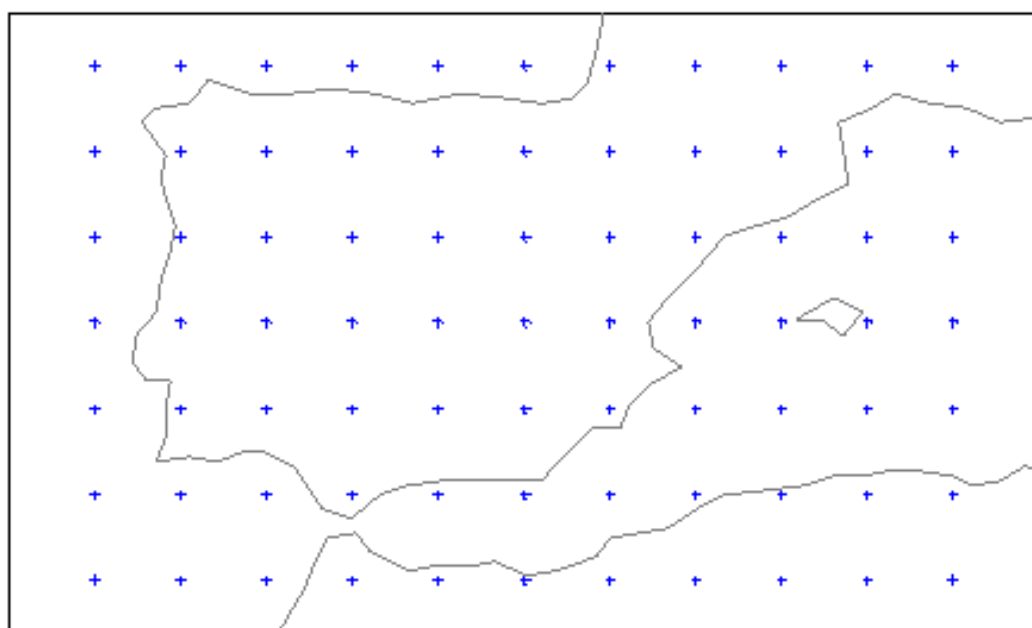


FIG 3.5 PATRÓN IBERIA 1.5°

Patrón navarra

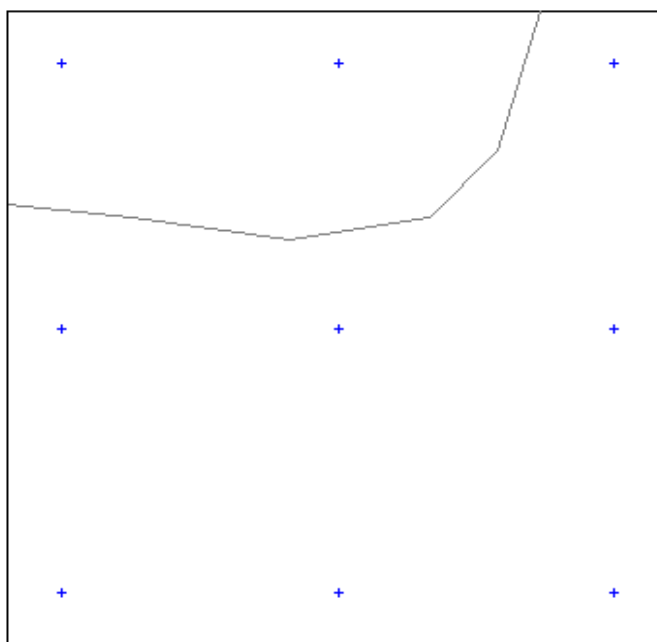


FIG 3.6 PATRÓN NAVARRA

Teniendo en cuenta esos cuatro modelos de rejillas se han definido los siguientes patrones para realizar las pruebas:

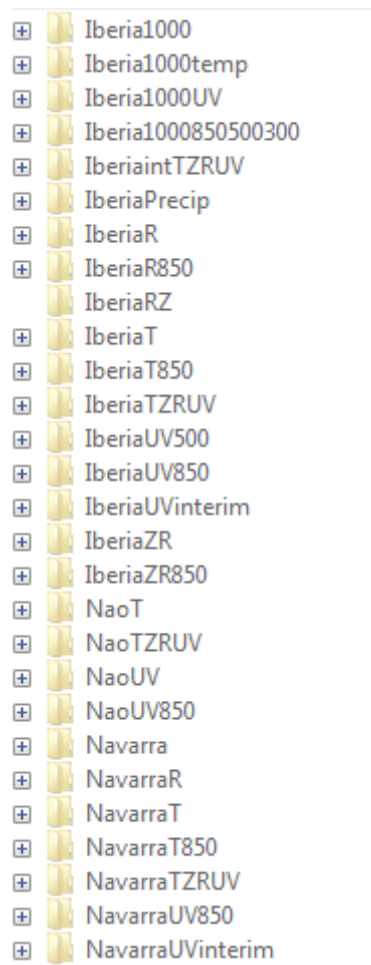


FIG 3.7 ESTRUCTURA DE PATRONES GENERADOS

Iberia1000: Con resolución 1.5° para las variables TZRUV a las 00 horas a 1000hPa.

Iberia1000temp: Con resolución 1.5° para las variables TZRUV a las 00, 06,12 y18 horas a 1000hPa.

Iberia1000UV: Con resolución 1.5° para las variables UV a las 00, 06, 12 y18 a 1000hPa

Iberia1000850500300: Con resolución 1.5° para las variables TZRUV a las 00, 06, 12 y18 horas a 1000, 850,500 y 300 hPa.

IbeiraintTZRUV: Con resolución 1.5° para las variables TZRUV a las 00 horas a 850 y 500hPa.

Iberiaprecip: Con resolución de 1° específico para lluvia (no se ha utilizado más que para una comprobación).

IberiaR: Con resolución 1.5° para la variable R a las 00 horas a 850 y 500hPa.

IberiaR850: Con resolución 1.5° para la variable R a las 00 horas a 850hPa.

IberiaRZ: Con resolución 1.5° para las variables R y Z a las 00 horas a 850 y 500hPa.

IberiaT: Con resolución 1.5° para la variable T a las 00 horas a 850h y 500hPa.

IberiaT850: Con resolución 1.5° para la variable T a las 00 horas a 850hPa.

IberiaTZRUV: Con resolución 1° (patrón ya computados los componentes principales).

IberiaUV500: Con resolución 1.5° para las variables U y V a las 00 horas a 500hPa.

IberiaUV850: Con resolución 1.5° para las variables U y V a las 00 horas a 850hPa.

IberiaUVinterim: Con resolución 1.5° para las variables U y V a las 00 horas a 850 y 500hPa

IberiaZR: Con resolución 1.5° para las variables R y Z a las 00 horas a 850 y 500hPa.

IberiaZR850: Con resolución 1.5° para las variables R y Z a las 00 horas a 850hPa.

NaoT: Con resolución 1.5° para la variable T a las 00 horas a 850hPa.

NaoTZRUV: Con resolución 1.5° para las variables TZRUV a las 00 horas a 850h y 500hPa.

NaoUV: Con resolución 1.5° para las variables U y V a las 00 horas a 850 y 500hPa.

NaoUV850: Con resolución 1.5° para las variables U y V a las 00 horas a 850hPa.

Navarra: Con resolución 1.5° para las variables TZRUV a las 00 horas a 850h y 500hPa.

NavarraR: Con resolución 1.5° para la variable R a las 00 horas a 850 y 500hPa.

NavarraT: Con resolución 1.5° para la variable T a las 00 horas a 850h y 500hPa.

NavarraT850: Con resolución 1.5° para la variable T a las 00 horas a 850hPa.

NavarraTZRUV: Con resolución 1.5° para las variables TZRUV a las 00 horas a 850h y 500hPa.

NavarraUV850: Con resolución 1.5° para las variables U y V a las 00 horas a 850hPa.

NavarraUVinterim: Con resolución 1.5° para las variables U y V a las 00 horas a 850 y 500hPa

Estos patrones se encuentran en la carpeta Iberiapatterns de la herramienta y se definen mediante un fichero de texto del siguiente tipo:

```
src=$METEOLAB_HOME/../../NWPData/Ib_int_tempo/

lon=-10:1.5:5
lat=44:-1.5:35

lvl=1000
par=U,V,
tim=00,06,12,18,

startDate=01-Jan-1989
endDate=31-Dec-2009
```

Es necesario poner la ruta del archivo GRIB del que se quieren obtener los datos, la longitud y latitud de la rejilla de donde se obtienen las variables, el nivel de presión del que se quieren obtener las variables, las variables que se quieren obtener, la hora de la salida y las fechas de los días.

Teniendo definidas de esa manera los patrones se pueden leer los ficheros GRIB y extraer las variables, calcular los CPs y representarlos con la toolbox meteolab [26].

Una vez que se tienen tanto los datos históricos de los patrones como los datos diarios ya se pueden realizar las predicciones.

3.4-DOWNSCALING

El problema denominado *downscaling* (o aumento de resolución) consiste en “interpolar” las predicciones de un modelo numérico dadas en una cierta rejilla a puntos sub-rejilla (una red de estaciones distribuidas irregularmente, una rejilla de mayor resolución, etc.).

En la práctica, este problema aparece cuando la resolución espacial del modelo numérico de circulación atmosférica (ACM) es insuficiente para una aplicación dada, o cuando se desean obtener predicciones locales en ciudades, o localidades de interés. El problema es aún más complicado cuando se desean obtener predicciones de variables en superficie cuya salida no proporcione directamente el modelo (temperatura máxima diaria, etc), sino que han de estimarse a partir de otras variables disponibles.

En el caso concreto de este proyecto, se utiliza la técnica de *downscaling* estadístico que se aplica cuando se dispone de registros históricos de la variable que se desea predecir en la rejilla de alta resolución, o en los puntos locales de interés.

En este caso, distintos métodos Estadísticos permiten combinar las salidas de los modelos numéricos con esta información, para desarrollar modelos de predicción que relacionen las salidas de los modelos numéricos con las observaciones locales.

Para ello, es necesario disponer de un conjunto suficiente de datos simultáneos de registros históricos y de campos atmosféricos simulados por algún proyecto de re-análisis

3.5-TIPOS DE PREDICCIÓN Y VALIDACIONES.

A la hora de realizar predicciones sobre eventos meteorológicos, se pueden tratar de diferentes formas, podemos encontrarnos con eventos binarios, como podría ser la presencia de nieve o no; categóricas, viento fuerte, moderado o débil; continuas como la temperatura o mixtas.

Además, algunas variables se pueden tratar de manera distinta según interese en el momento de estudio.

3.5.1-PREDICCIONES DETERMINISTAS

Las predicciones deterministas son aquellas en las que se proporciona el estado de la variables discreta, por ejemplo precipitación=15mm.

Algunos sistemas de predicción determinista elementales, utilizados como sistemas de referencia para validar otros métodos, son:

La persistencia, que consiste en pronosticar para un instante t lo que ocurrió en el instante $t - 1$ (por ejemplo, la precipitación predicha para hoy será la que se haya observado ayer):

$$\hat{o}(t) = o(t - 1)$$

También se puede considerar el caso más general de la persistencia (τ), donde el valor de $o(t)$ se obtiene a partir de $\{o(t - 1), \dots, o(t - \tau)\}$.

La climatología, que consiste en pronosticar utilizando algún parámetro derivado de la distribución climatológica obtenida empíricamente para un evento dado. Por ejemplo, la precipitación predicha para un día de Enero será la media de los valores diarios de precipitación en Enero en los últimos diez años.

3.5.2-PREDICCIONES PROBABILÍSTICAS

Una predicción probabilística describe en términos cuantitativos la incertidumbre asociada a la predicción. El pronóstico realizado es una distribución de probabilidad sobre el total de la variable.

Por ejemplo, si la variable es continua y la predicción se supone Gaussiana, entonces una predicción probabilística vendría dada por su media y varianza; por otra parte, si la variable es discreta, entonces la predicción vendría dada por su función de probabilidad.

3.5.3-CALIDAD DE LAS PREDICCIONES

Hay una gran cantidad de aspectos a tener en cuenta a la hora de valorar una predicción y comprobar su calidad, sobre todo en los sistemas probabilísticos.

En función del tipo de probabilidad que utilizan (conjunta, marginal o condicionada), los índices de verificación se pueden dividir en tres grandes grupos Wilson (2001):

Los que hacen uso de la función de probabilidad conjunta para analizar la correspondencia entre observado y previsto. Ej: Bias, correlación, Skill...

Los que utilizan probabilidades condicionadas a valores concretos de predicción. Ej: fiabilidad, resolución, definición...

Los que utilizan probabilidades condicionadas a valores concretos de la observación. Ej: discriminación e incertidumbre.

A continuación se van a describir algunas de estas medidas:

- **Bias:** Se refiere a la concordancia entre la predicción media y la observación media. Se definen de tal modo que un sesgo positivo es una sobre estimación y un sesgo negativo es una subestimación.
- **Asociación:** Indica el grado de relación lineal entre la observación y la predicción. Ejemplos de esto son la covarianza y la correlación.
- **Precisión:** Relativa a la similitud entre el valor de la predicción y el valor real observado, se promedia sobre una muestra de parejas individuales de predicciones y observaciones. Ej: error absoluto medio, Brier score...
- **Habilidad:** Es la precisión de un sistema referida a otro que se toma como referencia. Un valor positivo indica que el sistema es más hábil que el de referencia.

Para el caso particular de predicciones asociadas a eventos binarios, existen dos medidas especiales que las determinan:

- **Fiabilidad:** Para que la predicción de un evento binario se a fiable, la probabilidad prevista y la observada deben coincidir lo máximo posible en todo el rango de valores de probabilidad.
- **Resolución:** Cuantifica la desviación de la predicción respecto de la predicción constante dada por la probabilidad climatológica del evento. La resolución se suele representar gráficamente mediante la función de probabilidad de la variable probabilidad predicha

3.5.4-VALIDACIÓN PREDICCIONES DETERMINISTAS

Cuando se realiza una predicción determinista, el valor es una estimación del valor de una variable continua y la calidad del sistema se mide con relativa facilidad mediante medidas de precisión y habilidad. Estas funciones son medidas globales de precisión, en el sentido de que sólo consideran la distribución conjunta, calculando la distancia entre datos previstos y observados.

1. La suma de los cuadrados de los errores (*Sum of Square Errors*, SSE), definida como:

$$\sum_{p=1}^a ||y_p - \hat{y}_y||^2$$

2. La raíz cuadrada del error cuadrático medio (*Root Mean Square Error*, RMSE) definida como:

$$\sqrt{\sum_{p=1}^a ||y_p - \hat{y}_y||^2 / n}$$

3. El error absoluto medio:

$$\text{abs}_{p=1, \dots, a} ||y_p - \hat{y}_y||$$

4. La correlación

$$\frac{< (o_i - < o_i >) (\hat{o}_i - < \hat{o}_i >) >}{\sigma(o) \sigma(\hat{o})}$$

3.5.5-VALIDACIÓN PREDICCIONES PROBABILÍSTICAS

Así como a las predicciones deterministas se aplican a variables continuas, las predicciones probabilísticas se suelen realizar sobre variables categóricas o categorizadas. Las medidas más utilizadas son las generalizaciones de las medidas deterministas.

Brier Score

Se trata del error medio calculado en unidades de probabilidad para la ocurrencia de las distintas categorías de la variable:

$$BS = < (p_i - o_i)^2 >,$$

donde $p_i = P(\hat{o}_i = si)$, y o_i es 1 si ocurre el evento y 0 en caso contrario. El valor de BS es 0 para una predicción perfecta. Si se quieren comparar las precisiones de dos sistemas, es conveniente normalizar El BS. BSP es el *brier score* de la predicción.

Brier Skill Score

Normalmente el *Skill Score* (SS) de un sistema de predicción, referido a un sistema estándar, por ejemplo la climatología, con probabilidades previstas, p_c se calcula:

$$SS = \frac{P(o = 1|p) - P(o = 1|p_c)}{1 - P(o = 1|p_c)}$$

Mide la mejora en sistemas probabilísticos con respecto a un sistema de referencia de predicción (usualmente uno a largo plazo o climatológico). El valor 0 indica que no aporta nada sobre el sistema de referencia, un 1 es el resultado perfecto.

3.5.6-PREDICCIONES BINARIAS PROBABILÍSTICAS

En meteorología es muy común transformar variables en binarias, en base a un cierto umbral crítico, por ejemplo $\text{precip} > 10\text{mm}$, viento $> 60\text{km/h}$... De la misma forma, dada una predicción probabilística para una variable binaria, $p_i = P(\hat{o}_i)$, se puede obtener una predicción determinista considerando un umbral u para la probabilidad.

Cuando la probabilidad de la predicción supere ese umbral se predecirá la ocurrencia del evento, en caso contrario la predicción será la no ocurrencia del evento. Se puede ir variando el umbral para hacer un sistema más arriesgado o más conservador a la hora de predecir el evento. En estos casos se tienen una tabla de contingencia con los valores asociados a los aciertos de ocurrencia α_u , falsas predicciones β_u , omisiones γ_u y aciertos de no ocurrencia δ_u que dependerán del umbral u .

Las curvas ROC y el valor económico de una predicción binaria probabilística describen de forma global las posibles tablas de contingencia en función de los distintos umbrales u . Para ello, utilizan dos índices HIR y FAR para valorar los aciertos y fallos del sistema.

Curvas Roc

Las curvas ROC se obtiene representando la tasa de falsas alarmas (FAR) frente a la tasa de aciertos (HIR) para los diferentes umbrales de probabilidad u .

Mide la habilidad de la predicción para discriminar entre dos alternativas posibles, es decir la resolución o pericia. No es sensible al bias, es decir, no dice nada acerca de la fiabilidad. Una predicción sesgada puede tener una buena resolución y producir una buena

curva ROC, lo que implica que se puede mejorar la predicción a través de una calibración. Por lo tanto, puede ser considerado como una medida de la utilidad potencial.

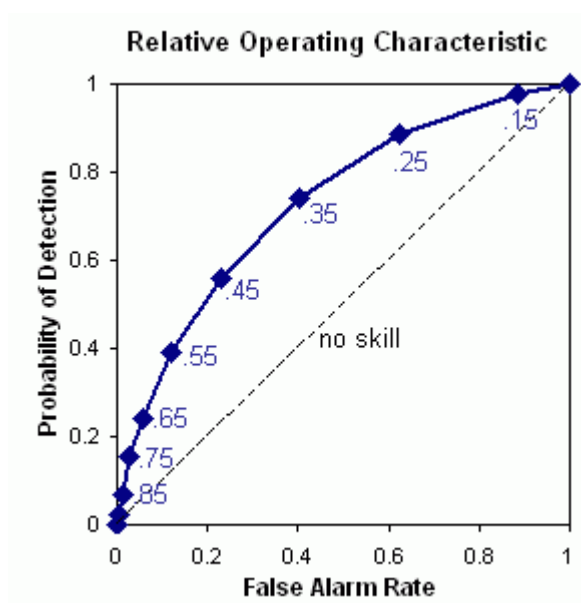


FIG 3.8 CURVA ROC

El área bajo la curva suele usarse también como un indicador.

La curva perfecta partiría del punto de más abajo ya la izquierda hasta el más alto y a la derecha del diagrama. La diagonal indica ninguna habilidad.

El área ROC varía entre 0 y 1, siendo 1 el mejor resultado y 0.5 el peor. Por lo tanto es un buen complemento para el diagrama de fiabilidad, que está condicionado a las previsiones.

Diagrama de resolución

El diagrama de fiabilidad representa la frecuencia observada frente a la probabilidad de la predicción, donde el rango de probabilidades de la predicción se divide en rangos. Proporciona información de que bien se corresponden las probabilidades previstas de un evento a las frecuencias observadas.

La fiabilidad se indica mediante la proximidad de la curva a la diagonal. La desviación de la diagonal proporciona el bias (sesgo) condicional. Si la curva está por debajo de la línea, indica un sobre-predicción y si es al revés, la curva por encima de la línea, indica sub-predicción.

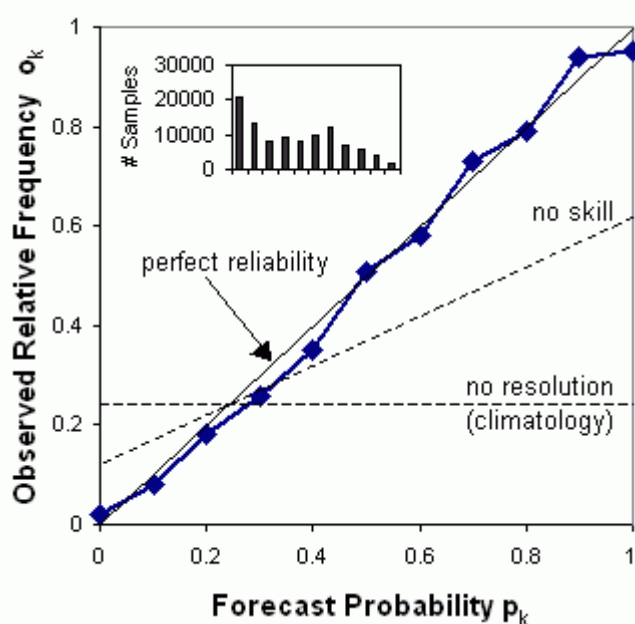


FIG 3.9 DIAGRAMA DE FIABILIDAD

Un pronóstico de la climatología no discrimina en absoluto entre los eventos y no-eventos, y por lo tanto no tiene fiabilidad. El diagrama de fiabilidad está condicionado a que las previsiones (por ejemplo, teniendo en cuenta que x se predijo, cual fue el resultado), y sirve para dar información sobre el verdadero significado de la previsión.

3.5.5-SCRIPTS

Una vez se han descrito los diferentes componentes para llevar a cabo las pruebas se van a detallar como se han realizado las diferentes *scripts* con los que se realiza el experimento:

Para ello se han utilizado estos scripts (adjuntados junto con todo el proyecto):

- Red_tempint.m: Predicción utilizando patrones de resolución 1.5° para la temperatura.
- Red_temp.m: Predicción utilizando el patrón de 1°.
- Red_precidet.m: Predicción determinista de precipitación normalizada.
- Red_precidet2.m: Predicción determinista de precipitación sin normalizar.
- Redpreciprob.m: Predicción probabilística de precipitación
- Redvientosprob.m: Predicción probabilística del viento.
- Redvientosvel.m: Predicción determinista del viento

- Dvientos.m: Predicción de la dirección del viento.

Para la regresión existe un único script que se llama regresión.m.

Ahora se va a detallar de manera más pormenorizada cómo son estos scripts mediante un ejemplo de cada tipo.

La Figura 3.10 muestra el *script* para la predicción determinista (en este caso de la temperatura):

```
%%Primero se carga el patrón
%dmn=readDomain('naoT');
%dmn=readDomain('iberiaintt');
%dmn=readDomain('navarraT');
%dmn=readDomain('navarraT850');
%dmn=readDomain('iberiaintt850');
%dmn=readDomain('iberia15');
%dmn=readDomain('iberia1000hpa');
%dmn=readDomain('iberiacomp');
dmn=readDomain('iberiatemp');

[EOF,CP]=getEOF(dmn,'ncp',50);

load 'tmax19892009.dat'
data=tmax19892009;

drawStations(dmn.nod');
pause();

j=find(isnan(data)==0); % Selecting days with no missing data
X=CP(j,:);
Y=data(j,:);

[X,PSx] = mapstd(X);
[Y,PSy] = mapstd(Y);

% [X,mX,dX,Y,mY,dY]=prestd(X,Y); % Data standarization
```

```

e=fix(size(X,2)*0.75); % Selecting 75 percent of data for training
Xt=X(:,1:e);
Yt=Y(:,1:e);

%nnet=newff([min(Xt,[],2),max(Xt,[],2)],[5 1],{'logsig','purelin'});
%nnet=newff([min(Xt,[],2),max(Xt,[],2)],[5 1],{'tansig','purelin'});
%nnet=newff([min(Xt,[],2),max(Xt,[],2)],[10 1],{'logsig','purelin'});
%nnet=newff([min(Xt,[],2),max(Xt,[],2)],[10 1],{'tansig','purelin'});
%nnet=newff([min(Xt,[],2),max(Xt,[],2)],[30 1]);
%nnet=newff([min(Xt,[],2),max(Xt,[],2)],[20 1],{'tansig','purelin'});

nnet=newff([min(Xt,[],2),max(Xt,[],2)],[10 5 1],{'tansig','tansig','purelin'});
%
% nnet=newff(Xt,Yt,[1]);

nnet.trainFcn='traingdx';
% nnet.trainFcn='traingdm';
% nnet.trainFcn='trainb';
nnet.trainParam.epochs=5000;
nnet=train(nnet,Xt,Yt);
P=sim(nnet,X);
% Pred=((P.*repmat(dY,[1 size(P,2)]))+repmat(mY,[1 size(P,2)]));
Pred = mapstd('reverse',P,PSy);
Obsr=data(j,:);

figure;
plot(Obsr(e+1:end,:), 'b');
hold on; plot(Pred(e+1:end,:), 'r');
figure
plot(Obsr,Pred, 'r')

%training error
mse(Y(1:e)-P(1:e))
%test error
mse(Y(e+1:end)-P(e+1:end))

%%Vamos a mostrar las diferencias en un vector

%diferencia1= abs(Obsr(e+1:end,:) - data(e+1:end,:));

diferencia= abs(Obsr(e+1:end,:) - Pred(e+1:end,:));

```

```
error_medio= sum(diferencia)/length(diferencia)
```

```
[n,m]=size(Obsr);
```

```
v=length(Obsr);
```

```
Obsr(v-100:v,1);
```

```
Pred(v-100:v,1);
```

```
Compara=[Obsr(v-100:v,1) Pred(v-100:v,1)]
```

FIG 3.10 SCRIPT DE PREDICCIÓN RED NEURONAL TEMPERATURA

El proceso que sigue el *script* es el siguiente:

1. Se lee el patrón con los datos atmosféricos con la función *readDomain()*.
2. Después se cogen los valores de los componentes principales previamente computados *getEOF()*.
3. Se leen los datos de la estación y se cargan en memoria.
4. Se comprueba los datos que faltan para que coincidan la pareja patrón-observación.
5. Se estandarizan los datos mediante la función de MATLAB *mapstd* o *prestd* para evitar un mal condicionamiento del problema.
6. Se selecciona una parte de los datos para entrenamiento y otra para test.
7. Se crea la red neuronal, se definen los parámetros y se entrena.
8. Se simula la red con los datos
9. Se vuelve a los valores auténticos previos a la estandarización.
10. Se muestran gráficos de valores observados vs previstos, se calculan los errores cuadráticos medios y se muestra una tabla con valores observados y previstos. También se calcula el error medio cometido.

En el caso de las previsiones probabilísticas hay que definir un umbral y convertir la predicción a binaria.

```
j=find(isnan(data)==0);
```

```
X=CP(j,:);
```

```
Y=data(j,:);
```



```
Y=Y>0.1;
```

FIG 3.11 CONVERSIÓN UMBRAL

Después se crea una red neuronal con salida “losig” que proporciona un valor entre 0 y 1 que será la probabilidad asociada a ese patrón.

```
%Resultados

Resbin=Pred>0.5;
Res = Resbin';

acierto = sum(abs(Y(:,e+1:end) - Res(:,e+1:end)))/length(Y(:,e+1:end));
porcentaje=(1-acierto)*100

aciertotot = sum(abs(Y - Res))/length(Y);
porcentajetot=(1-aciertotot)*100

compara=[Pred(e+1:end,:); objetivo(e+1:end,:)'];

val_det=validationBinaryDet(Y,Res);
```

FIG 3.12 VALORACIÓN ERROR PROBABILÍSTICA

Por otro lado los resultados se tratan de forma distinta y se ponen un umbral (en el caso del ejemplo es 0.5) con el que se considera que se produce el evento. Luego se calcula el porcentaje de acierto de test y el total con los valores entrenados.

Se ejecuta también la función de validación que nos proporcionará los índices y gráficos para la validación de predicciones binaria y probabilísticas.

Un caso un poco distinto es el de la clasificación de los vientos. En este caso se tienen cuatro clases de viento a clasificar y la salida son vectores con 5 componentes (el 5 indica que no hay predominancia de vientos), el índice en el que el valor sea mayor indica cuál es la clase.

Para calcularlo en el script se ve en la Figura 3.13:

```
for i=1:length(P)
    [c,m]=max(P(:,i));
    l(i)=m;
end
```

FIG 3.13 CÁLCULO MÁXIMO ÍNDICE

Y para contar los resultados iguales y calcular el porcentaje:

```
for i=1:length(Y)
    if Y(i)==I(i)
        cont=cont+1;
    end
end
cont;

(cont/length(Y))*100
```

FIG 3.14 CÁLCULO PORCENTAJES

El script de regresión es muy sencillo:

```
%%%REGRESION LINEA MÚLTIPLE

%dmn=readDomain('Nao');
% dmn=readDomain('Iberia');
% dmn=readDomain('iberiaintt');
% dmn=readDomain('Espintv');
% dmn=readDomain('iberiacomp');
% dmn=readDomain('navarraT');
dmn=readDomain('navarraUV850');
% dmn=readDomain('NaointerimTZRUUV');
% dmn=readDomain('iberiatemp');
% dmn=readDomain('navarraR');

% [EOF,CP]=getEOF(dmn,'ncp',50);
[EOF,CP]=getEOF(dmn);
% CP=patterns;

%%Se leen los datos de la estación%%%%%%%%%%%%%%

% load 'pamplonamax.dat';
% data = pamplonamax;

% load 'tmax19892009.dat'
% data=double(tmax19892009);

% load 'tmin19892009.dat'
% data=tmin19892009;
```

```

% data = pamplonatmin;

load 'pamplonavviento.dat';
data = pamplonavviento;

% load 'preci19892009.dat';
% data = preci19892009;

j=find(isnan(data)==0); % Selecting days with no missing data
X=[ones(size(j),1), CP(j,:)];
y=data(j,:);

c=regress(y,X);

%Forecasting using the regression

yhat=X*c;
figure
plot([y,yhat]);
xlabel('Dias');
ylabel('mm');
title('Observados vs previstos');
legend('Observados','Previstos');
figure;
plot(y,yhat,'k');
title('Observados vs previstos');
mse(y-yhat)

%%Vamos a mostrar las diferencias en un vector

diferencia= abs(y - yhat);

error_medio= sum(diferencia)/length(diferencia)

```

FIG 3.15 SCRIPT REGRESIÓN

Se cargan los datos de los patrones y de las estaciones de la misma manera que en los *scripts* de las redes neuronales y se utiliza el comando *regress*. En este caso no se normalizan los datos.

4- RESULTADOS EXPERIMENTALES

A continuación se van a mostrar algunos de los datos más representativos que se han obtenido al realizar las pruebas utilizando todo lo que se ha comentado en los capítulos anteriores. Se hace un mayor énfasis en las pruebas con redes neuronales aunque también hay pruebas para la regresión.

4.1-REDES NEURONALES

En esta primera parte se detallan los resultados para las pruebas realizadas con diferentes estructuras de redes neuronales con diferentes parámetros y usando diferentes patrones.

4.1.1-TEMPERATURA

Patrón Iberia completo con 50 CP con monocapa para la máxima:

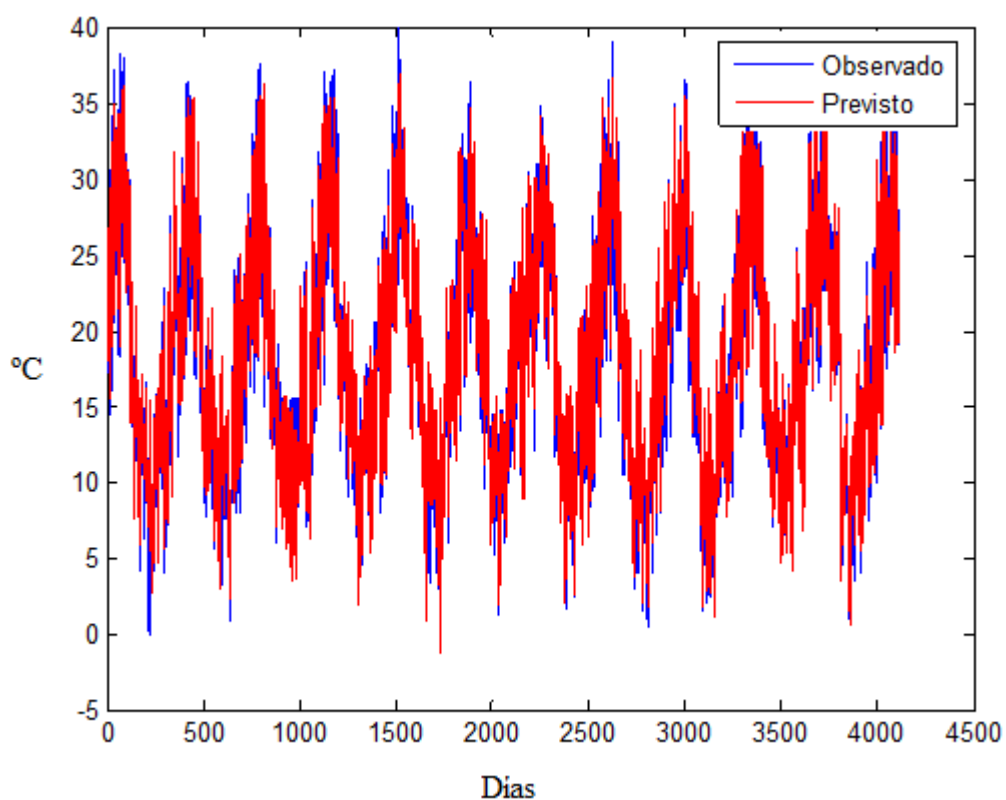


FIG 4.1 OBSERVADOS VS PRECISTOS PATRÓN IBERIA

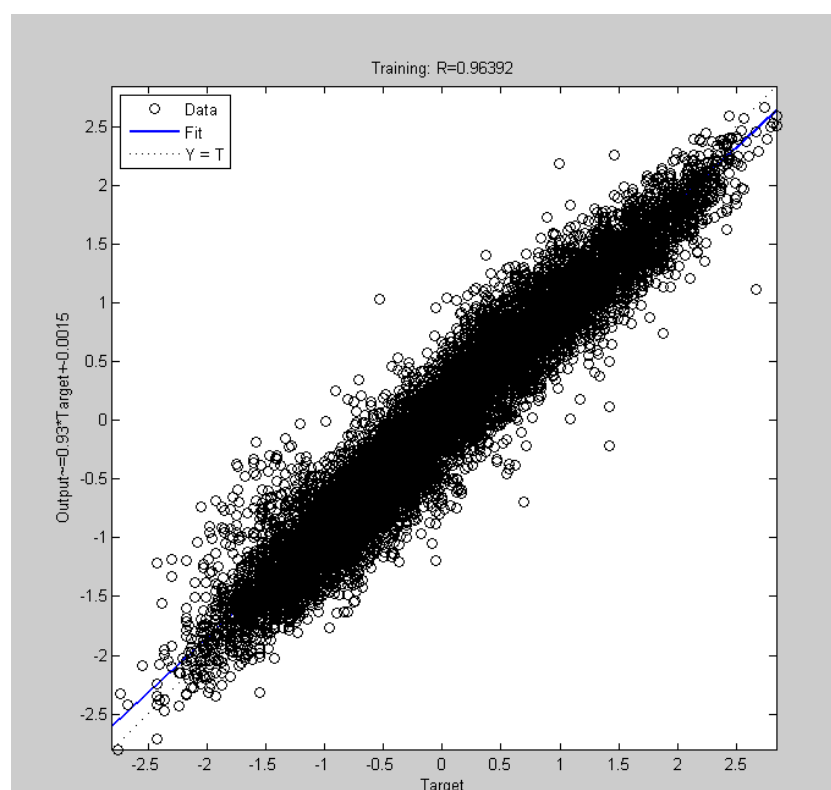


FIG 4.2 OBSERVADOS VS PRECISTOS REGRESIÓN

Error entrenamiento	Error test	Error medio
0.0711	0.0677	1.5984

OBSERVADO	PREVISTO		OBSERVADO	PREVISTO
15	14,6003		25	26,504
24	21,7519		19	19,2402
18	17,2596		20,5	20,3265
18	16,9127		19,5	21,2935
16,5	15,2761		21	22,4708
23	20,7702		24,5	24,9789
20	20,5508		32	30,5043
27,5	24,6837		35,5	34,7765
31	29,7335		35,5	33,7167
29	28,0735		31,5	30,5485
32,5	29,5363		31,5	29,2788
25,5	25,4585		21	23,8661
17	22,3197		21	24,0912
13,5	14,3523		24,5	24,1706
18	13,6995		26	27,1402
16,5	19,1044		35,5	31,9999

14,5	13,7117	33,5	30,7375
19	16,8929	33	29,9428
18,5	18,5799	23	23,9649
26,5	24,7324	21,5	21,4943
31,5	28,6979	23,5	21,6713
33,5	30,9451	28	27,0284
33,5	34,4051	27	27,6613
34,5	31,8411	30	27,2997
35	33,8264	28,5	27,777
35,5	35,7344	22	23,1311
27,5	30,1591	22,5	23,3629
30	28,732	22	21,6274
25,5	27,5372	22	20,9129
32	30,4073	19,5	18,4681
37	33,2554	20,5	20,138
25,5	29,9598	19,5	23,7673
18,5	22,9891	26	27,6101
26	27,946	35	32,9017
33,5	31,3455	31	29,9447
33,5	28,632	32,5	29,6051
19	22,5169	34	31,6842
19,5	21,4871	36	31,4916
17,5	18,9111	35	29,7831
27,5	24,096	27,5	26,7786
21	23,3294	28,5	27,5777
27	25,979	25	25,1576
21	22,6662	25	23,9768
25	25,5088	24,5	21,3821
20	21,251	21,5	20,2088
32,5	28,8997	20,5	19,3944
25,5	28,4434	20,5	18,9905
28,5	25,0935	21,5	19,3882
26	22,2114	19	22,1543
24	25,1559	28	26,5066

TABLA RESULTADOS ALEATORIOS DE PREDICCIÓN PATRONES NO USADOS EN
ENTRENAMIENTO

Mínima:

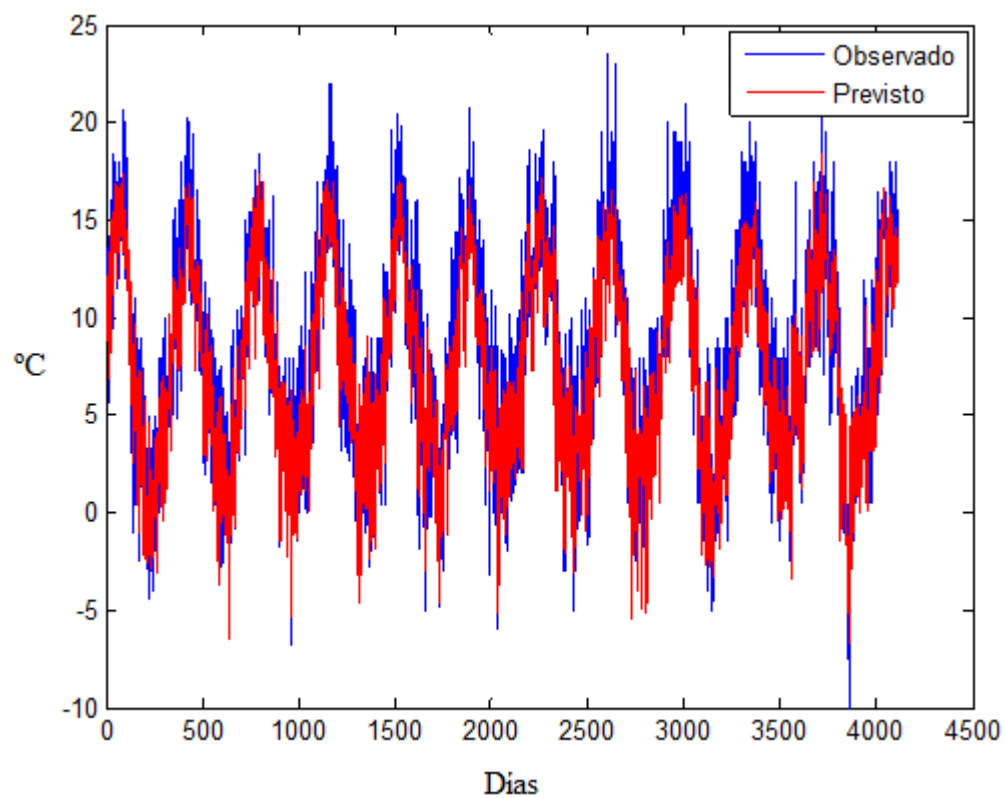


FIG 4.3 OBSERVADOS VS PRECISTOS PATRÓN IBERIA MÍNIMA

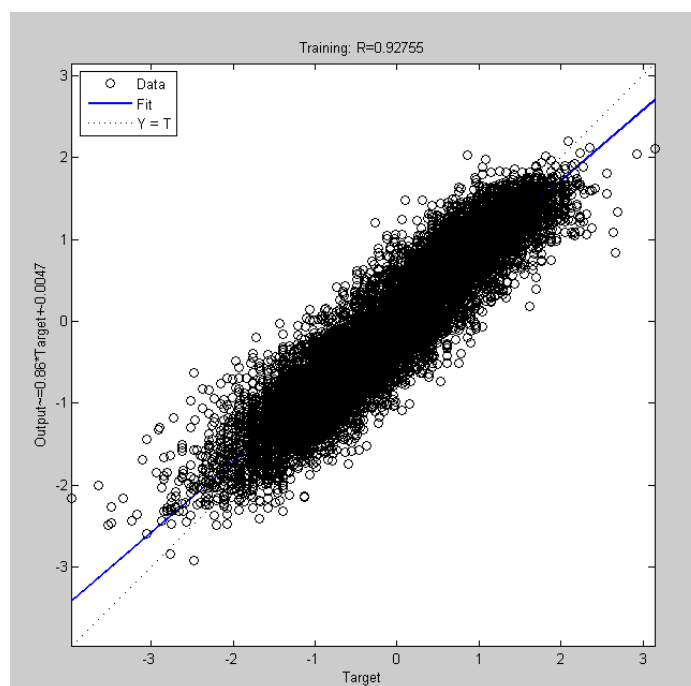


FIG 4.4 OBSERVADOS VS PREVISTOS REGRESIÓN

Error entrenamiento	Error test	Error medio
0.1388	0.1519	1.7276

OBSERVADO	PREVISTO		OBSERVADO	PREVISTO
8,5	7,0388		8	11,7239
6	10,3098		14,5	13,0733
11	7,9514		8	12,7037
8,5	7,0388		15	15,1135
6	10,3098		15	13,8837
11	7,9514		11	13,0559
6	7,2454		13	12,9744
8	6,1489		10	14,5159
7,5	7,4843		13	11,1377
10,5	9,8099		12,5	12,2818
10	11,7705		13,5	12,7298
14	12,7315		13,5	12,7522
15	14,3423		12	12,4743
14	14,4648		12	13,1042
14	13,3161		11,5	14,9616
12,5	12,083		18	16,2706
10	7,7767		16	15,6737
7	7,7773		15,5	15,8206
7	9,6916		15	15,2176
8	7,363		13	13,803
8,5	7,9164		15,5	15,2359
10,5	9,505		16,5	15,7276
7	11,2917		13	14,8087
11	12,3422		15,5	15,1565
13,5	13,5035		17	15,2943
16	14,0363		17,5	15,3747
14,5	14,3112		13,5	12,0321
15,5	15,377		10	10,3695
16	15,4639		9,5	11,2646
16,5	15,0197		12,5	12,8178
15,5	15,9764		11,5	13,6587
14	16,308		15	14,3597
14,5	15,3854		14,5	14,6387
16,5	16,0271		14	12,7978
16,5	16,676		15	12,7225
14,5	15,6787		13	11,1488
15	14,9382		11,5	10,418

14	14,7511	11,5	11,328
16,5	14,0267	12	14,2122
13,5	13,9483	13	11,7471
11,5	12,5867	11	14,1079
12,5	10,2358	16	13,9675
8,5	11,8855	15	13,6119
14	13,8037	16	14,0856
9,5	13,0759	17	13,6123
13	13,3222	16,5	14,6125

Para una red con una capa oculta de 5 neuronas y utilizando entrenamiento traingdx:

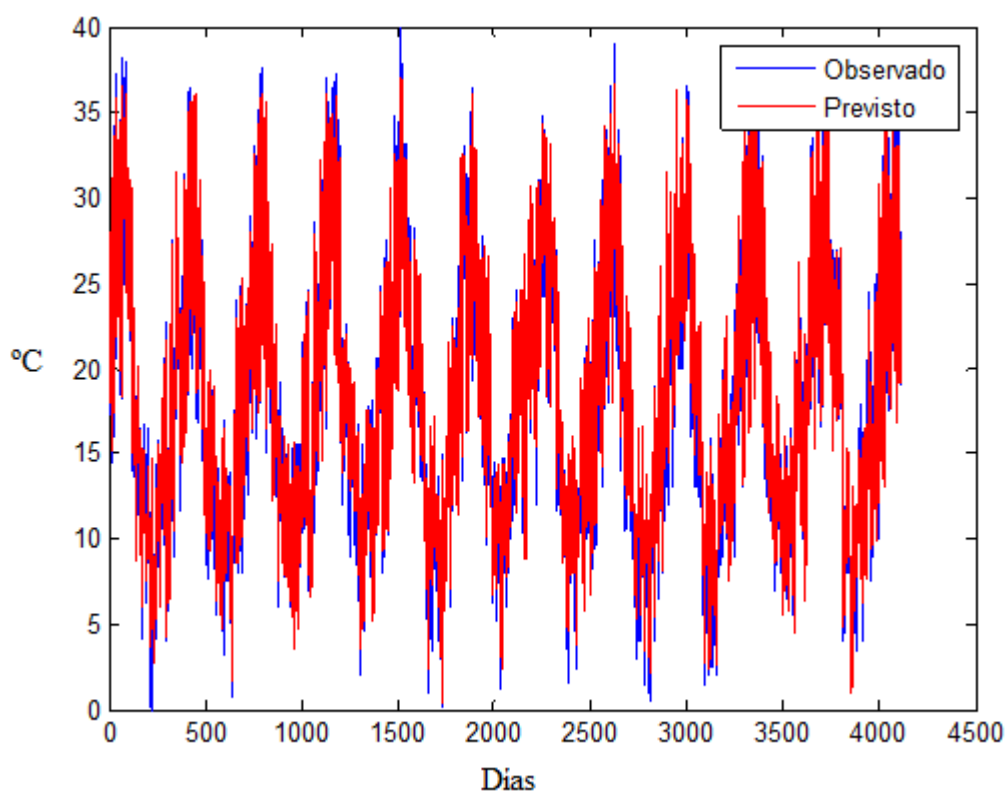


FIG 4.5 OBSERVADOS VS PREVISTOS PATRÓN IBERIA

Error entrenamiento	Error test	Error medio
0.0505	0.0510	1.3709

Utilizando la misma red pero entrenamiento levenberg-mardquart:

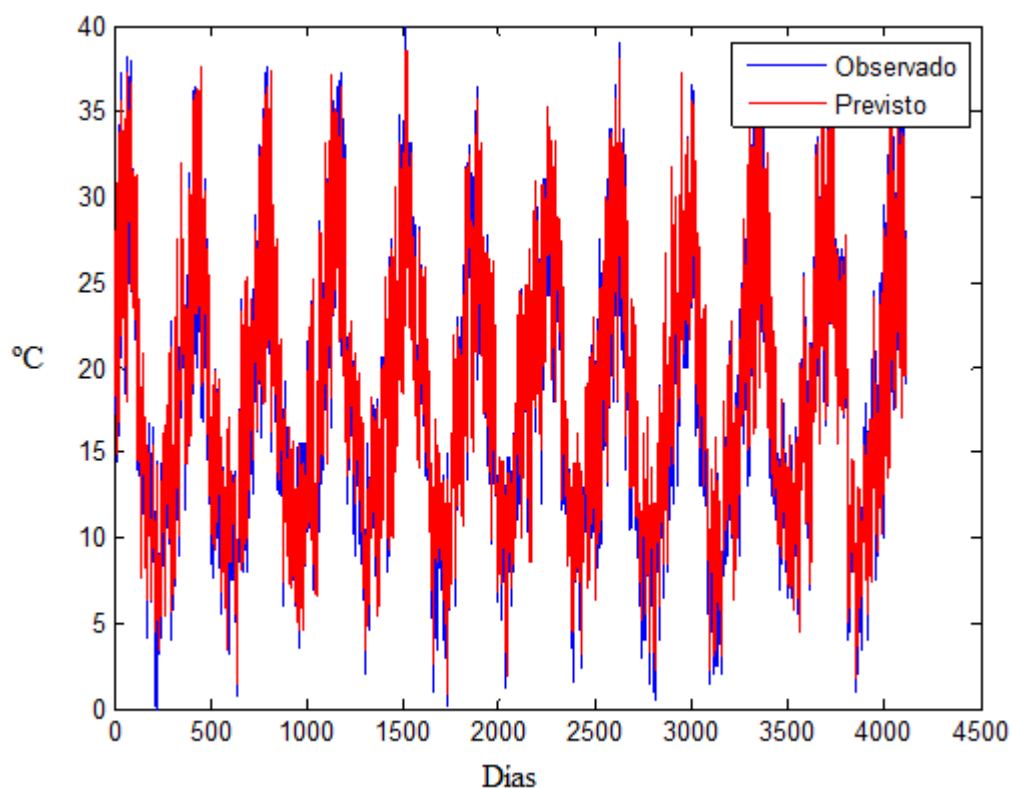


FIG 4.6 OBSERVADOS VS PREVISTOS IBERIA “LEVENBERG-MARQUARDT”

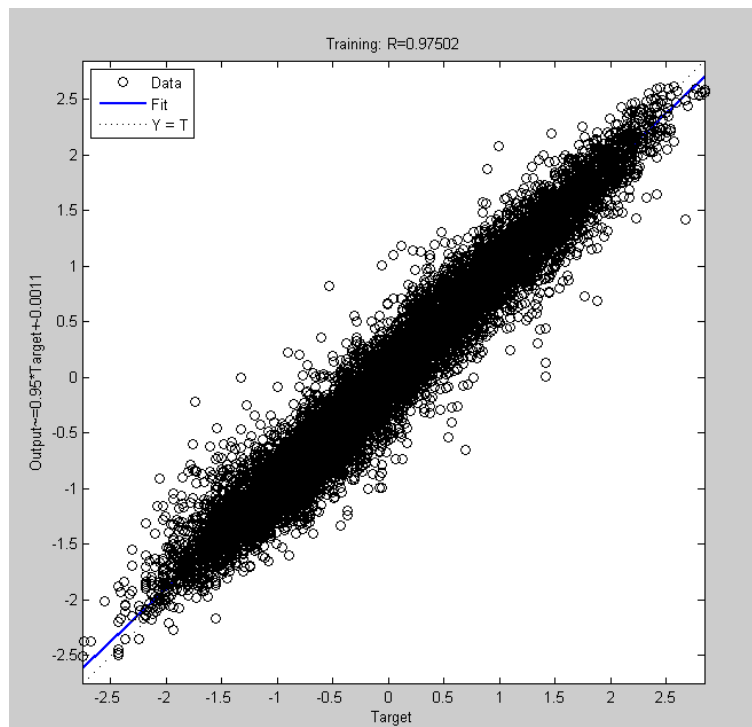


FIG 4.7 OBSERVADOS VS PREVISTOS IBERIA “LEVENBERG-MARQUARDT” REGRESIÓN

Error entrenamiento	Error test	Error medio
0.0495	0.0533	1.4115

Para la mínima

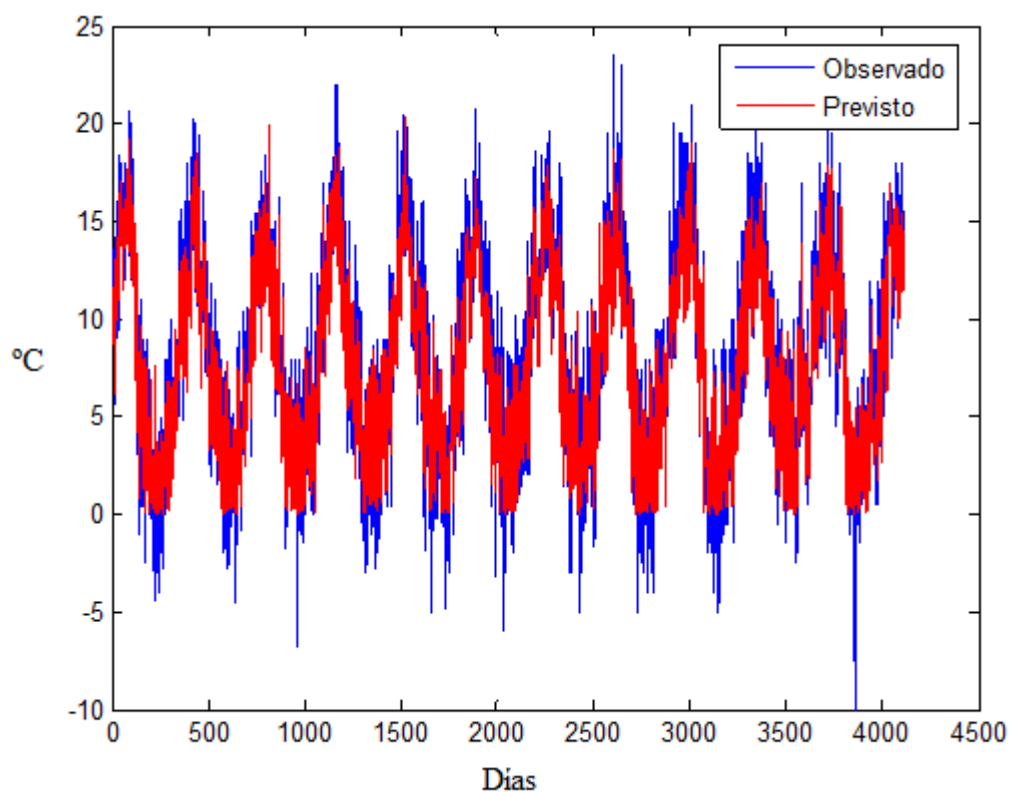


FIG 4.8 OBSERVADOS VS PREVISTOS IBERIA

Error entrenamiento	Error test	Error medio
0.1073	0.1310	1.5913

Se prueba también con salida tansig en vez de logisg y los resultados son peores:

Error entrenamiento	Error test	Error medio
0.1106	0.1316	1.6087

Red neuronal con una capa oculta con 10 neuronas

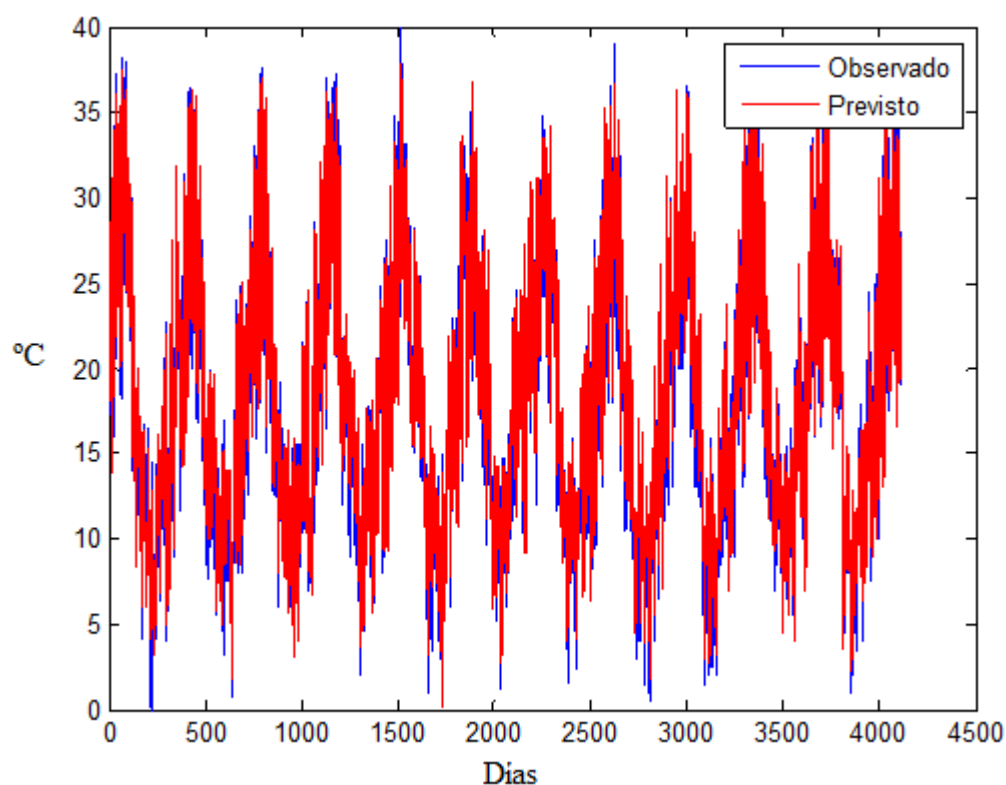


FIG 4.9 OBSERVADOS VS PREVISTOS IBERIA

Error entrenamiento	Error test	Error medio
0.1106	0.1316	1.6087

Error entrenamiento	Error test	Error medio
0.0466	0.0519	1.3876

Cambiando el entrenamiento a traingdm los resultados son peores:

Error entrenamiento	Error test	Error medio
0.0688	0.0639	1.5568

También utilizando trainb:

Error entrenamiento	Error test	Error medio
0.0611	0.0604	1.5165

Para la mínima con Levenberg-Marquardt:

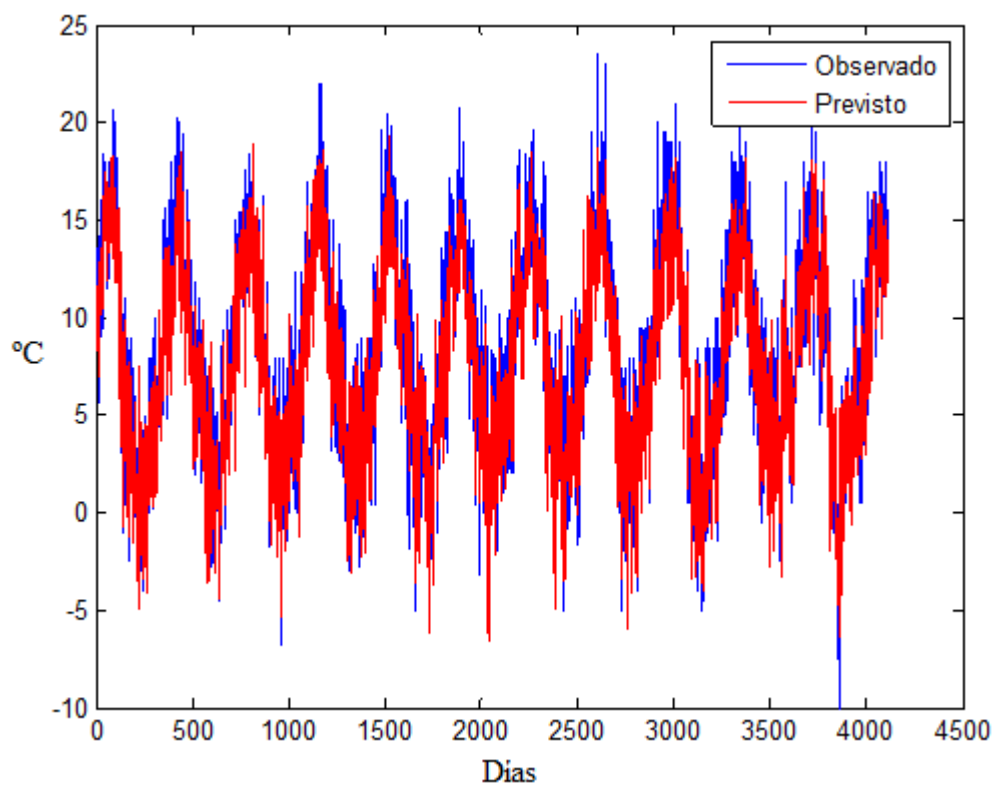


FIG 4.10 OBSERVADOS VS PREVISTOS IBERIA

Error entrenamiento	Error test	Error medio
0.0952	0.1262	1.5562

Con traingdx:

Error entrenamiento	Error test	Error medio
0.0962	0.1245	1.5430

Igual pero con tansig:

Error entrenamiento	Error test	Error medio
0.0962	0.1230	1.5388

Cambiando el entrenamiento a traingdm también en este caso los resultados son peores:

Error entrenamiento	Error test	Error medio
0.1020	0.1240	1.5452

También utilizando trainb:

Error entrenamiento	Error test	Error medio
0.1002	0.1369	1.6319

Aumentamos el número de neuronas a 20 en la capa oculta

Máxima

Error entrenamiento	Error test	Error medio
0.0401	0.0544	1.4283

Error entrenamiento	Error test	Error medio
0.0389	0.0604	1.5047

Mínima

Error entrenamiento	Error test	Error medio
0.0840	0.1400	1.6454

Error entrenamiento	Error test	Error medio
0.0847	0.1320	1.5811

Se observa que no hay mejora al aumentar el número de neuronas de la capa oculta, mejora un poco el error de entrenamiento pero tanto el error de test como el error medio son mayores.

Utilizando una red con 2 capas ocultas:

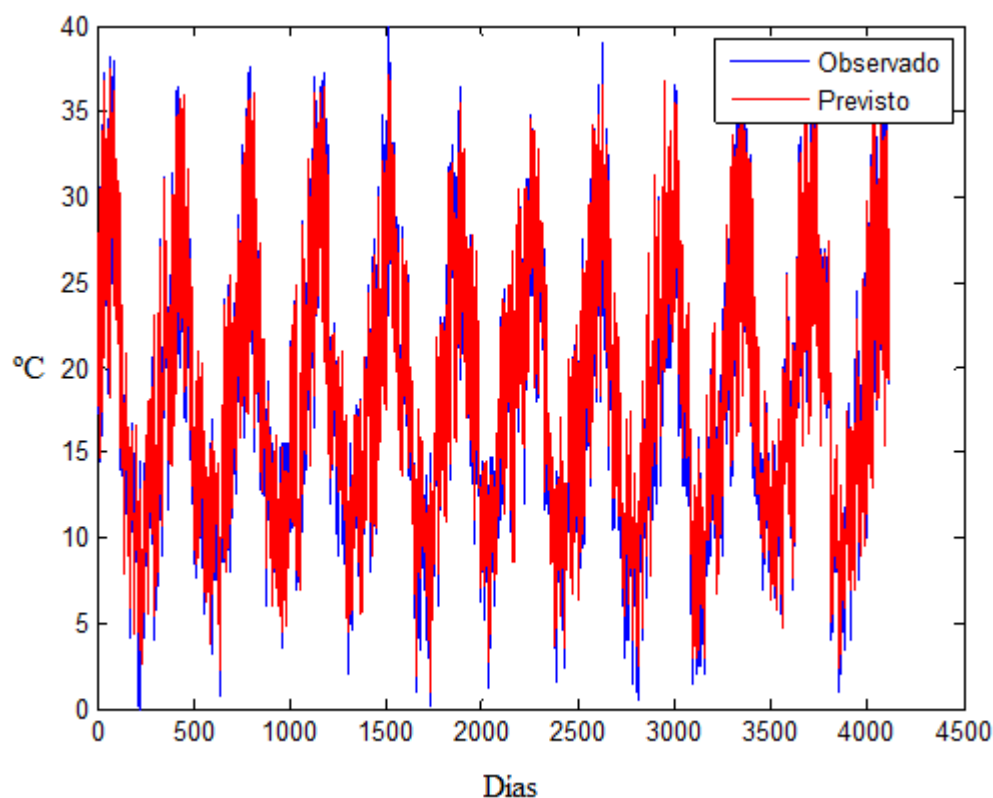


FIG 4.11 OBSERVADOS VS PREVISTOS IBERIA

Con 8 y 5 neuronas respectivamente en las capas ocultas.

Levenberg-marquardt

Error entrenamiento	Error test	Error medio
0.0471	0.0530	1.4126

Traingdx

Error entrenamiento	Error test	Error medio
0.0463	0.0496	1.3649

Si se aumenta el número de neuronas tampoco mejora mucho los resultados, con 30 y 15 respectivamente:

Error entrenamiento	Error test	Error medio
0.0384	0.0540	1.4329

Consideramos el patrón de temperatura a 850hPa y500hPa en Nao, Navarra y Iberia con 1,5° de resolución.

Red con una capa oculta de 5 neuronas y traingdx para el entrenamiento:

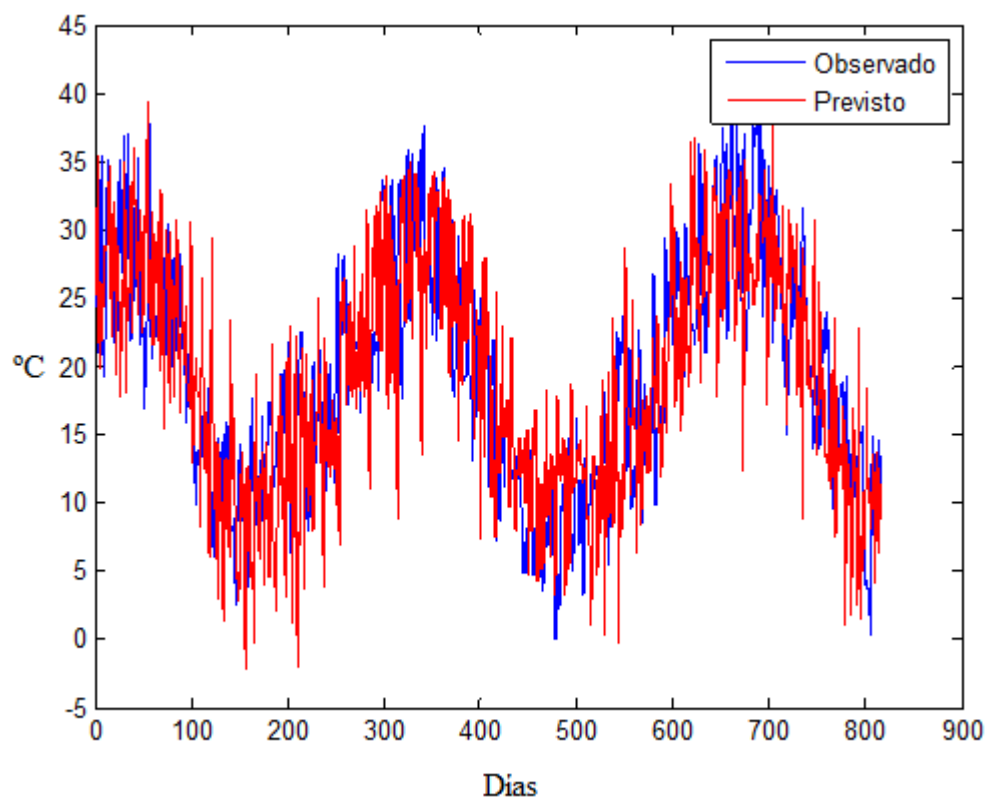


FIG 4.12 OBSERVADOS VS PREVISTOS NAO

Error entrenamiento	Error test	Error medio
0.100	0.6532	5.4264

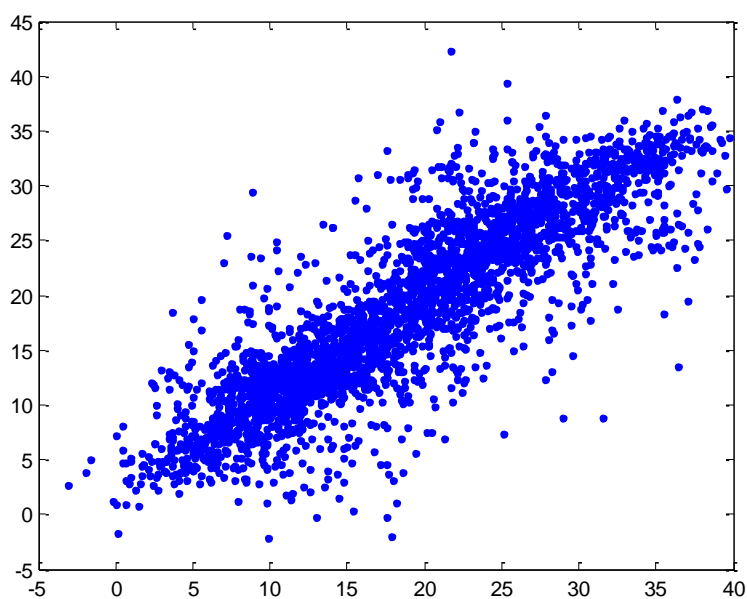


FIG 4.13 OBSERVADOS VS PREVISTOS NAO REGRESIÓN

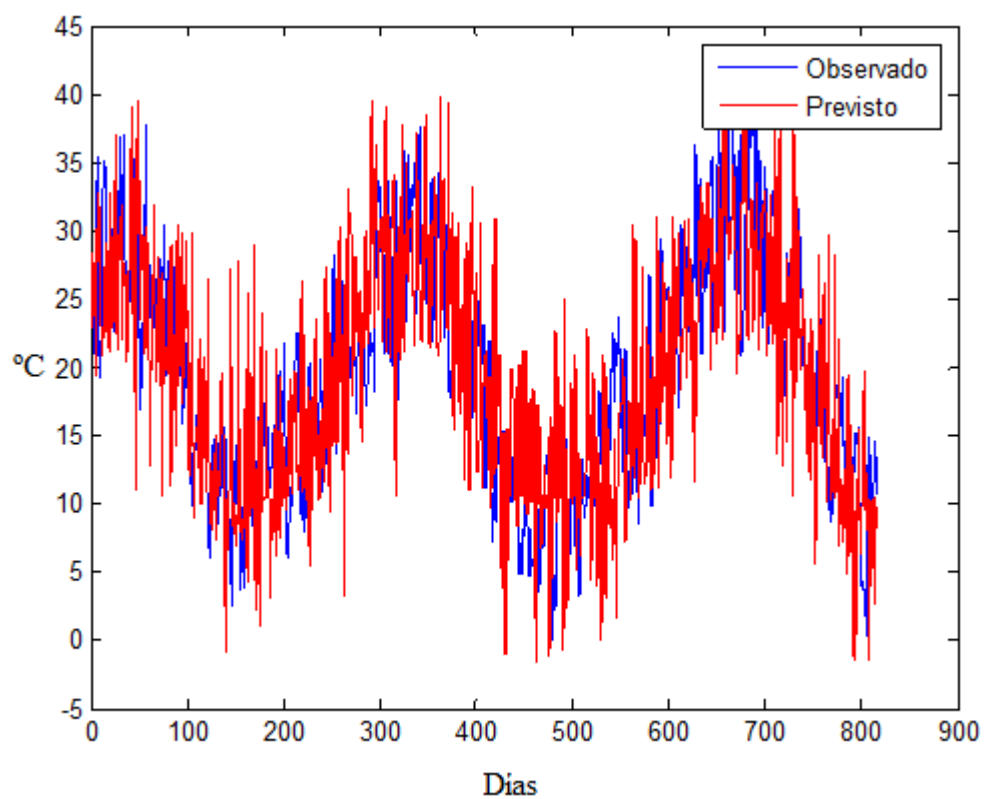
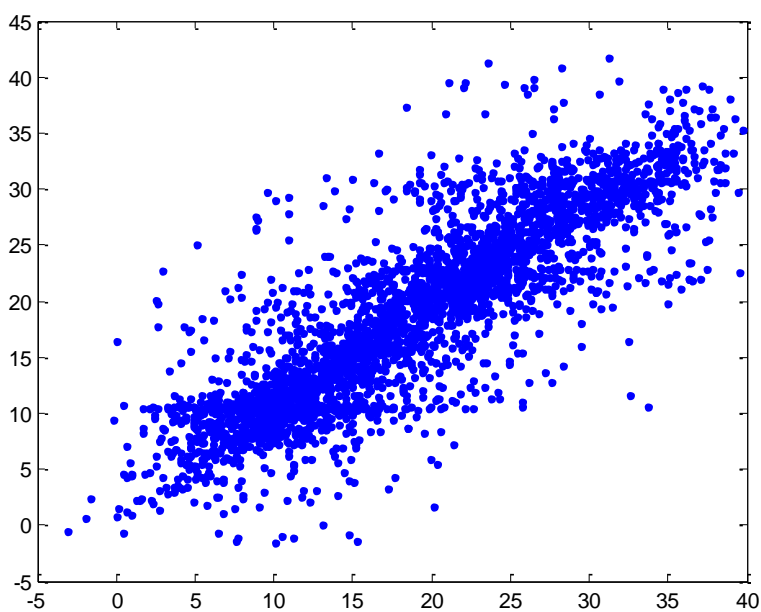


FIG 4.14 OBSERVADOS VS PREVISTOS IBERIA 1.5°

Error entrenamiento	Error test	Error medio
0.1143	0.7706	6.0153



5.15 OBSERVADOS VS PREVISTOS IBERIA1.5 REGRESIÓN

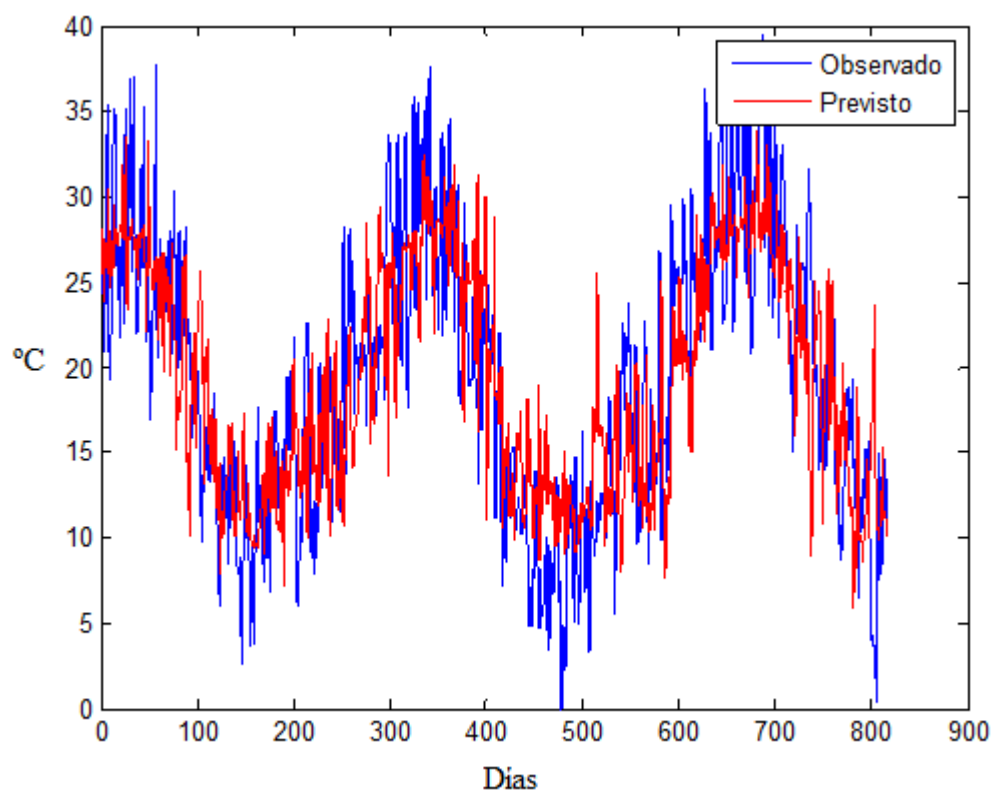


FIG 4.16 OBSERVADOS VS PREVISTOS NAVARRA 1.5°

Error entrenamiento	Error test	Error medio
0.3659	0.4512	4.5688

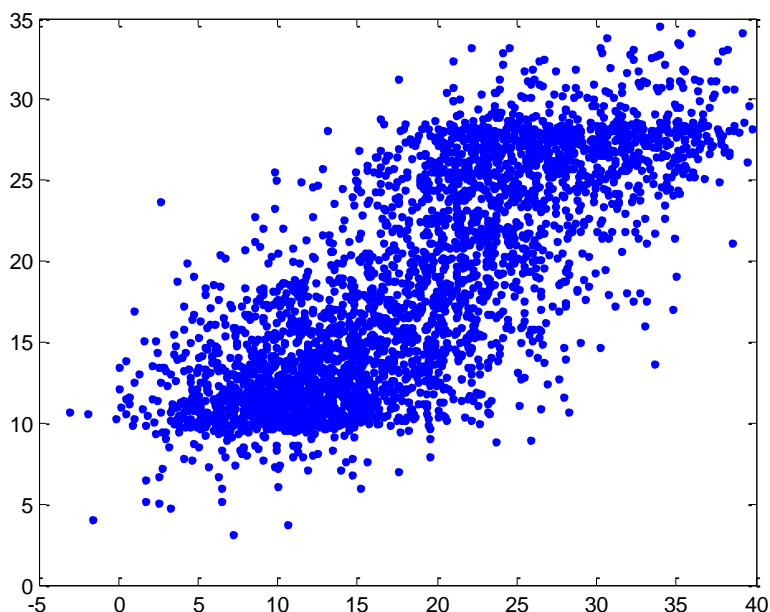


FIG 4.17 OBSERVADOS VS PREVISTOS NAVARRA 1.5° REGRESIÓN

Como se observa los resultados son peores que con el patrón de mayor resolución.

Si se opta por poner más neuronas (10) y más componentes principales:

Error entrenamiento	Error test	Error medio
1.8322e-018	0.8678	6.3123

Se observa que el error de entrenamiento es bajísimo pero la red no aprende bien, dando peores resultados incluso.

Si se opta por dos capas ocultas con 10 y 5 neuronas, para el patrón de iberiaT tampoco mejoran los resultados:

Error entrenamiento	Error test	Error medio
0.0140	1.5286	8.4466

Los patrones más específicos no funcionan mejor que el de mayor resolución.

Para el patrón con componentes temporales iberiatemp:

Error entrenamiento	Error test	Error medio
0.0347	0.0601	1.5817

Para iberia1000hPa:

Error entrenamiento	Error test	Error medio
0.0683	0.0969	2.0096

4.1.2-PRECIPITACIÓN

Con una red monocapa y patrón Iberia se puede ver que los resultados no son muy buenos:

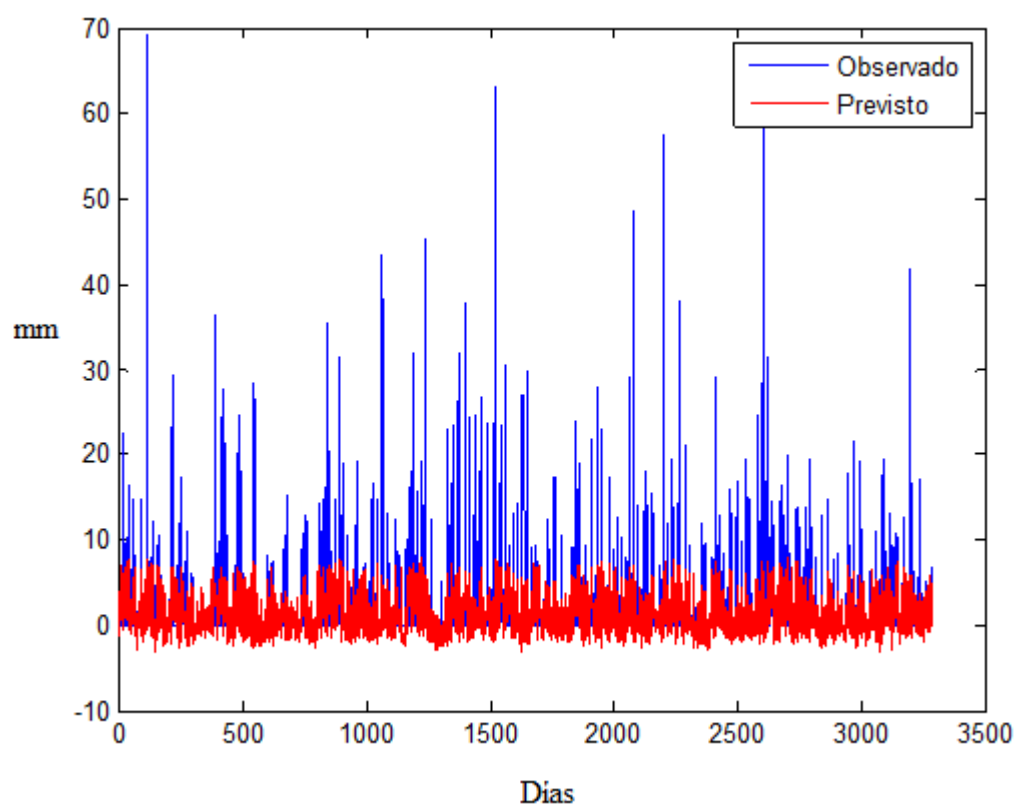


FIG 4.18 OBSERVADOS VS PREVISTOS IBERIA

Para una red con una capa oculta:

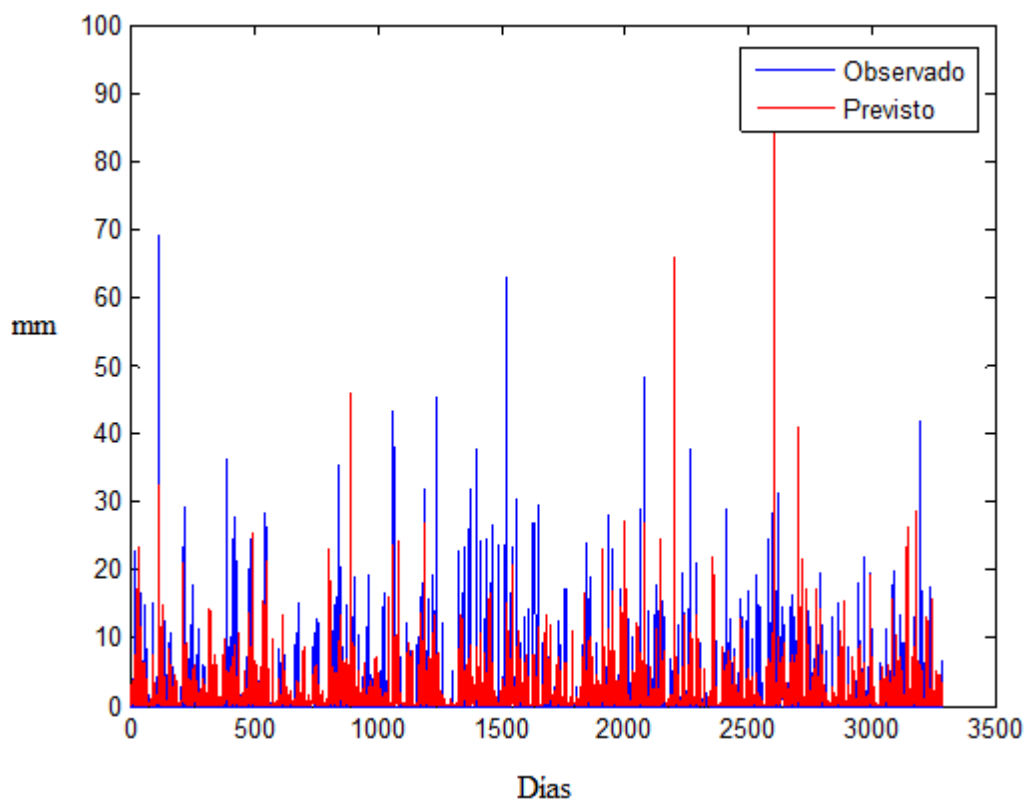


FIG 4.19 OBSERVADOS VS PREVISTOS 1 CAPA OCULTA

Error entrenamiento	Error test	Error medio
0.5188	0.5772	2.0266

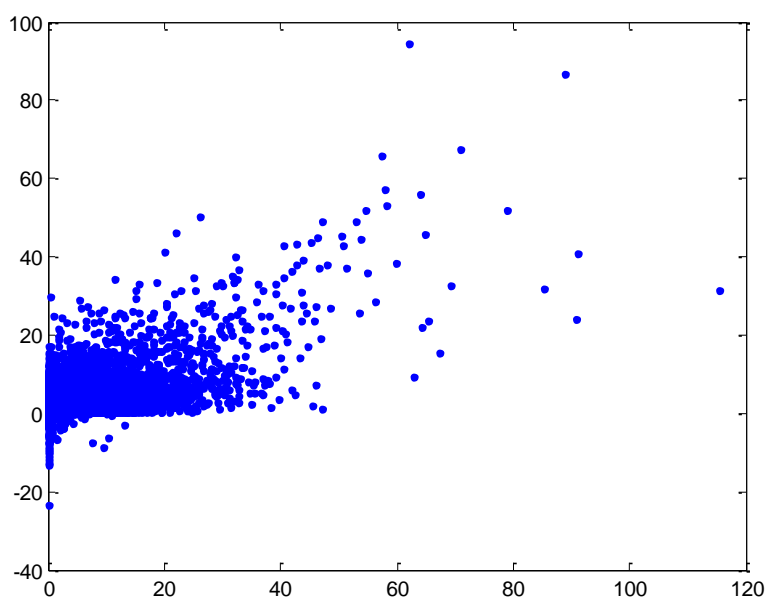


FIG 4.20 OBSERVADOS VS PREVISTOS REGRESIÓN IBERIA

Con una capa oculta traingdx y otro patrón (iberiacomp).

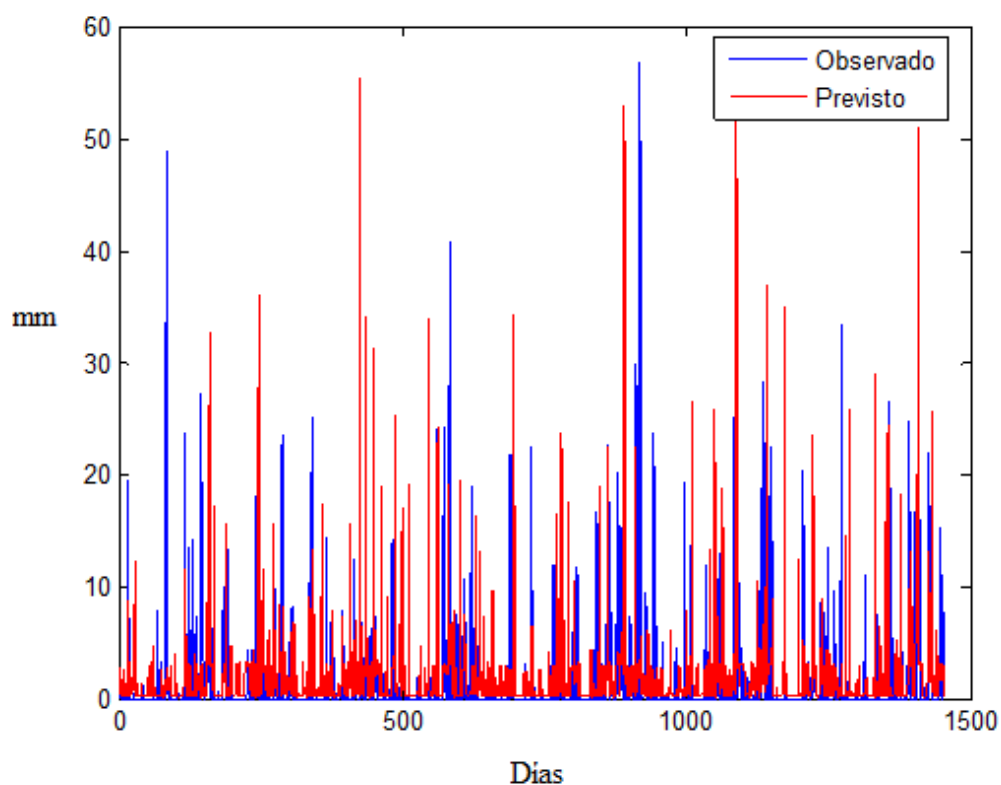


FIG 4.21 OBSERVADOS VS PREVISTOS IBERIACOMP 1CAPA

Error entrenamiento	Error test	Error medio
0.3547	1.5796	2.8173

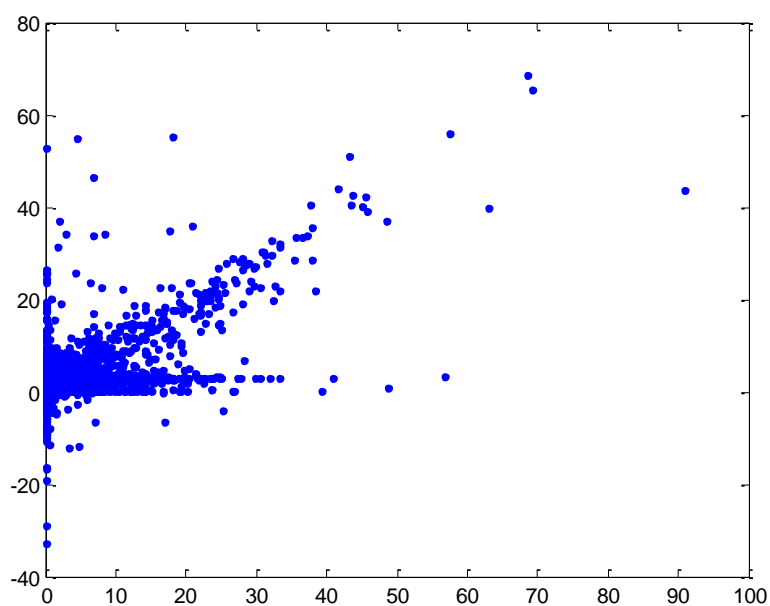


FIG 4.22 OBSERVADOS VS PREVISTOS REGRESIÓN

Se prueba con dos capas oculta y el patrón iberia a 1°

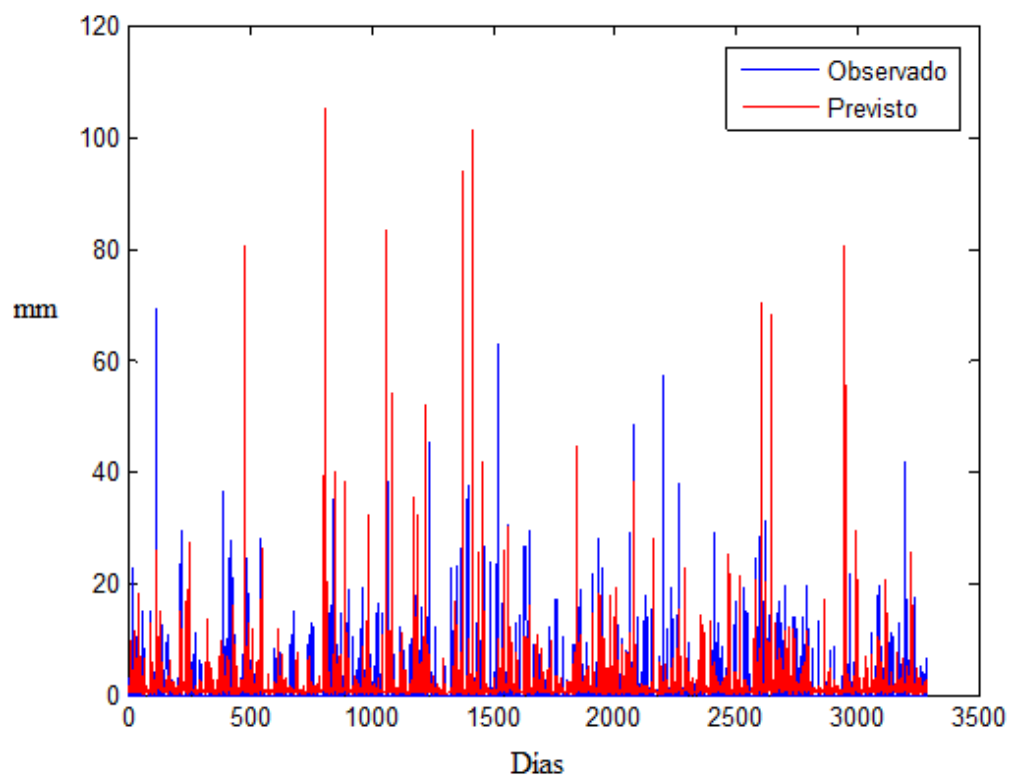


FIG 4.23 OBSERVADOS VS PREVISTOS IBERIA 2 CAPAS

Error entrenamiento	Error test	Error medio
0.3036	1.0302	2.4399

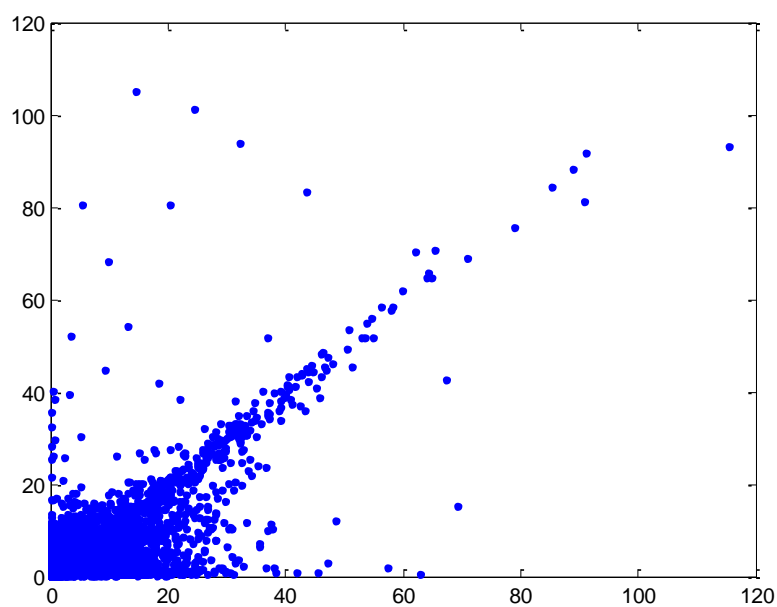


FIG 4.24 OBSERVADOS VS PREVISTOS IBERIA 2 CAPAS REGRESIÓN

Ahora con un patrón más específico a 1000hpa solamente también con dos capas ocultas

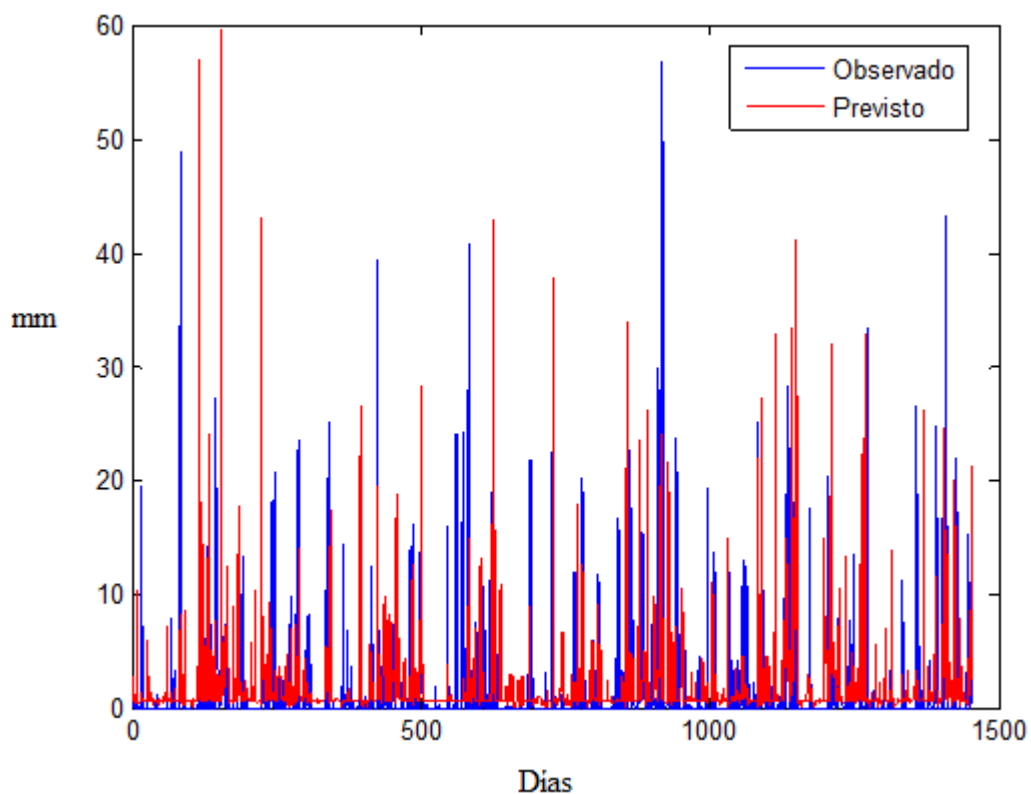


FIG 4.25 LLUVIA PREVISTA VS OBSERVADA EN MM DOS CAPAS OCULTA Y EL PATRÓN IBERIA A 1°

Error entrenamiento	Error test	Error medio
0.3036	1.0302	2.4399

Tampoco se mejora el resultado.

Considerando un patrón con recorrido temporal sobre 1000hPa

Error entrenamiento	Error test	Error medio
0.1283	2.4656	2.6805

Con patrones más específicos considerando la humedad relativa y el geopotencial:

Para el geopotencial y temperatura a unos 850 hPa, con una capa oculta 5 neuronas

Error entrenamiento	Error test	Error medio
0.1993	2.5473	3.0014

Error entrenamiento	Error test	Error medio
0.4962	1.5338	2.6367

Hacer una predicción determinista de la lluvia es mucho más complicado que la temperatura ya que la distribución estadística es mucho más irregular que en el caso de la temperatura. Los fallos que se cometen a la hora de predecir son mayores. Aunque el error medio no es muy grande no se predicen bien las lluvias.

Una predicción probabilística de la precipitación nos da menos detalles pero una mejor predicción y fiabilidad. En las pruebas que se realizan para obtener el mejor resultado se va a considerar un umbral del 50%. Se usan 50 CP.

Con un patrón de la NAO, y una capa oculta con 20 neuronas:

Error entrenamiento	BSS	Porcentaje
0.155	0.4053	79.7031
BSC	BSP	RSA
0.2268	0.1349	0.7358

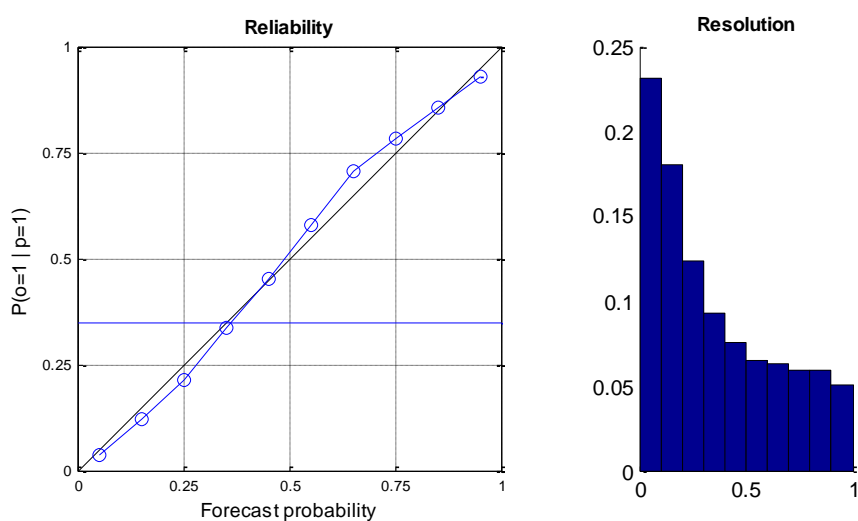


FIG 4.26 FIABILIDAD NAO 1CAPA OCULTA

Con una red con dos capas ocultas para el mismo patrón

Error entrenamiento	BSS	Porcentaje
0.131	0.4098	79.9221
BSC	BSP	RSA
0.2268	0.1339	0.7412

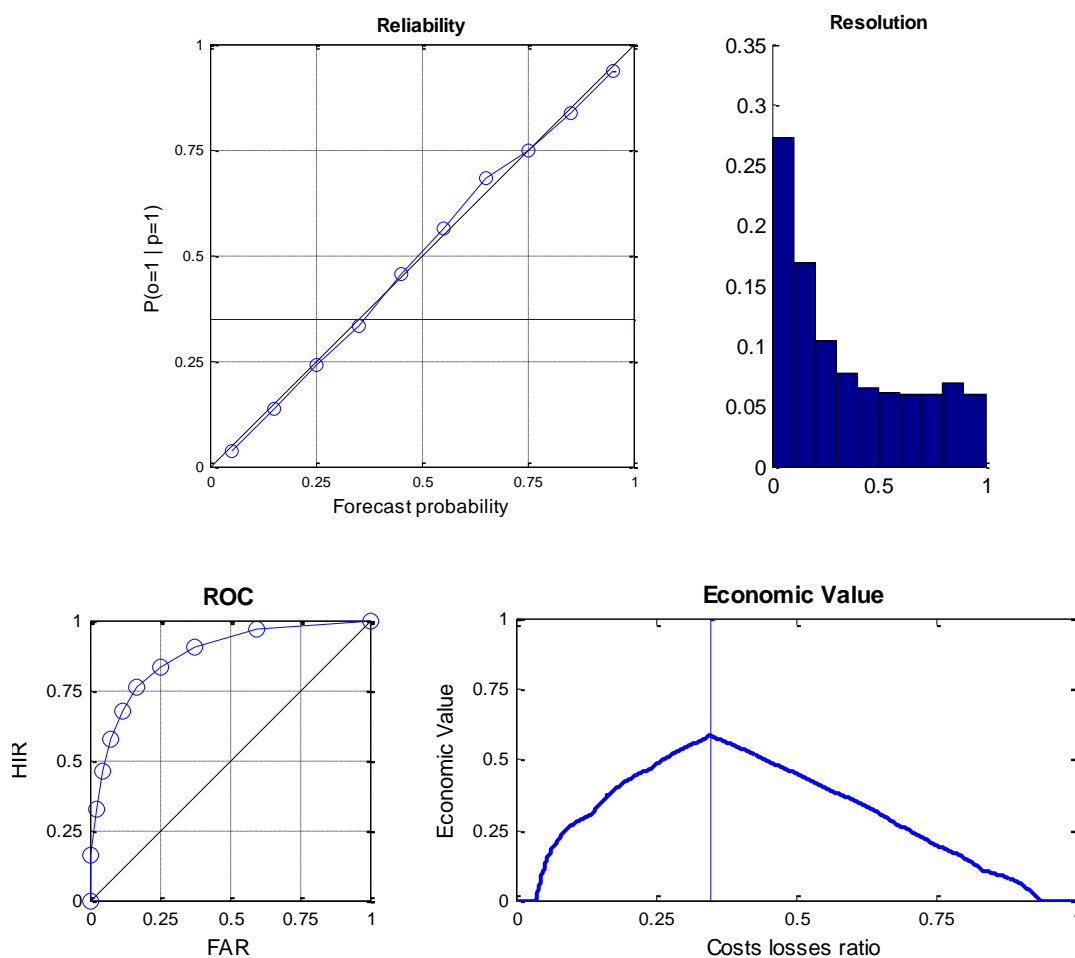


FIG 4.27 GRÁFICOS VALIDACIÓN 2 CAPAS OCULTAS NAO

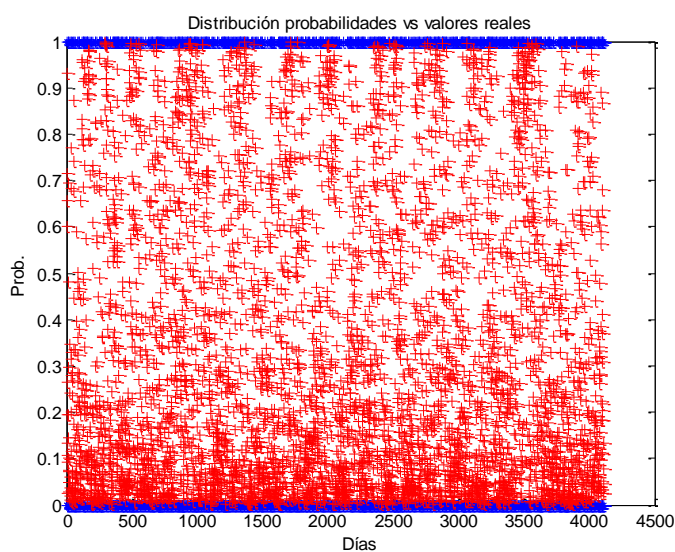


FIG 4.28 CLASIFICACIÓN 2 CAPAS OCULTAS NAO

La misma prueba con la función de entrenamiento de Levenberg-Marquardt:

Error entrenamiento	BSS	Porcentaje
0.132	0.4102	79.8004
BSC	BSP	RSA
0.2268	0.1338	0.7390

Con el patrón iberia de 1° de resolución y con traingdx:

Error entrenamiento	BSS	Porcentaje
0.109	0.5050	82.3071
BSC	BSP	RSA
0.2268	0.1123	0.8072

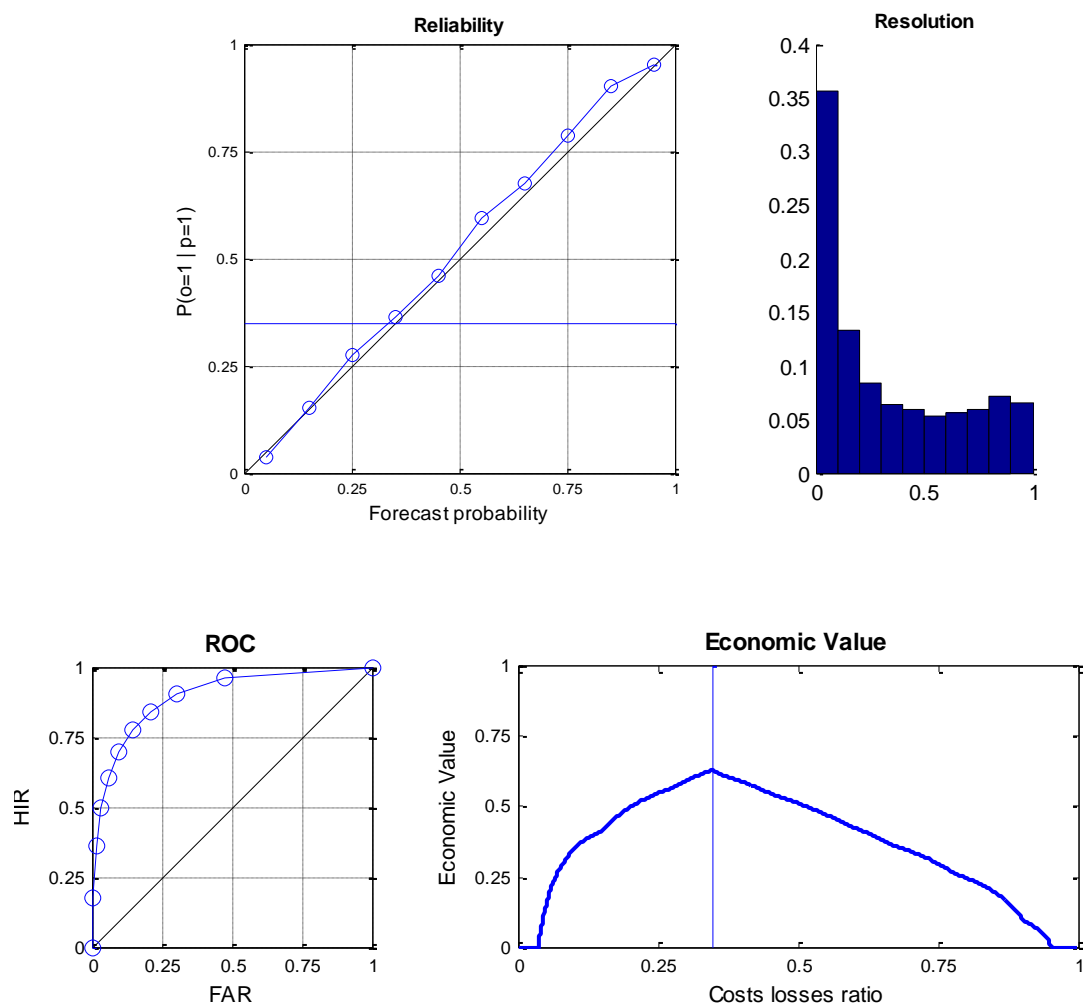


FIG 4.29 GRÁFICOS VALIDACIÓN IBERIA 2 CAPAS

Se ve que este patrón da mejores resultados que el anterior.

Un modelo similar pero de 1.5° de resolución:

Error entrenamiento	BSS	Porcentaje
0.109	0.4326	78.0099
BSC	BSP	RSA
0.2227	0.1264	0.7479

Con componentes temporales sobre 1000hPa:

Error entrenamiento	BSS	Porcentaje
0.0942	0.5683	83.2325
BSC	BSP	RSA
0.2227	0.0961	0.8382

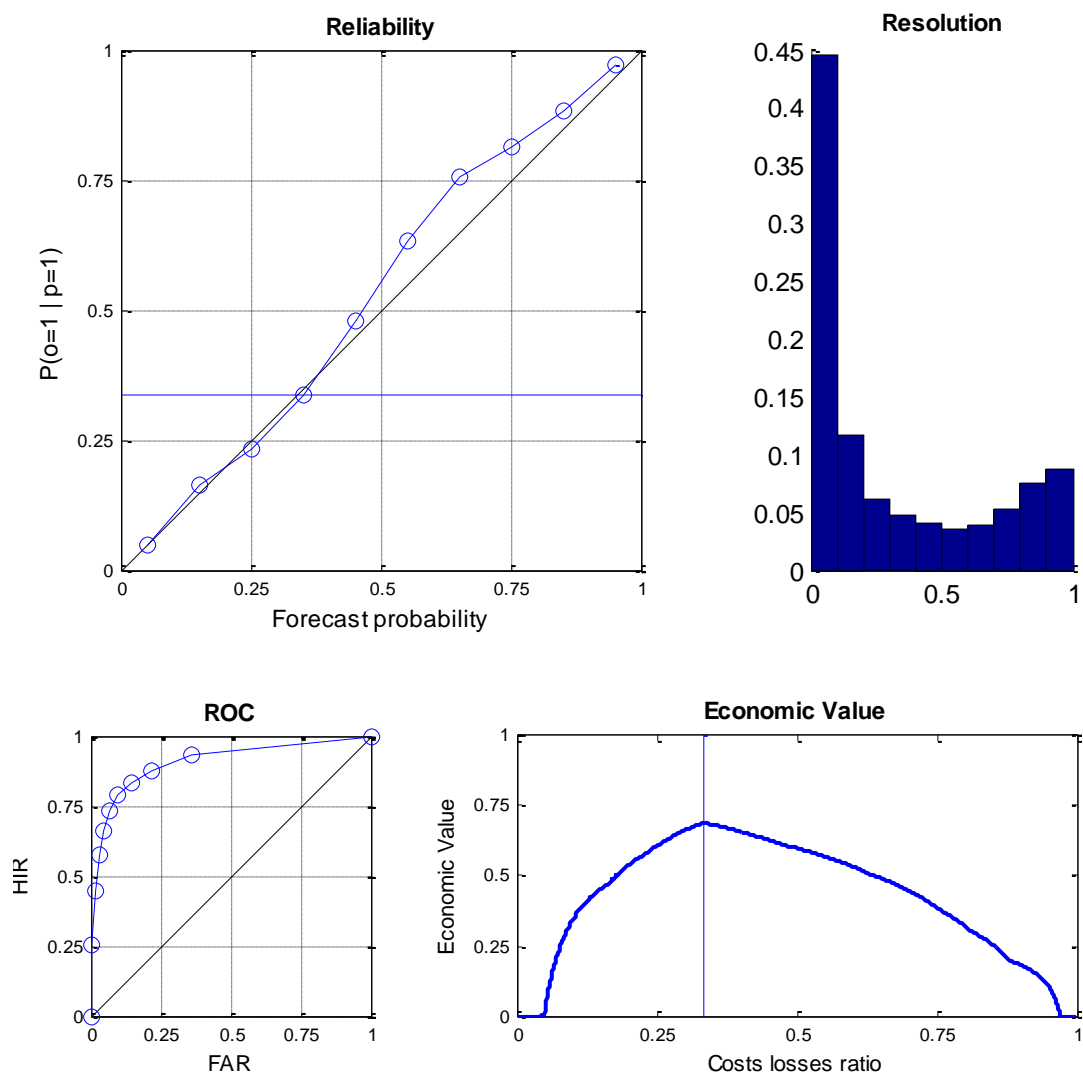


FIG 4.30 GRÁFICOS VALIDACIÓN IBERIATEMP

Para patrones más específicos, teniendo en cuenta la humedad relativa y el geopotencial con los mismos parámetros en la red neuronal que nos ha dado el mejor resultado:

IberiaRZ

Error entrenamiento	BSS	Porcentaje
0.121	0.4202	77.8450
BSC	BSP	RSA
0.2227	0.1291	0.7388

IberiaR

Error entrenamiento	BSS	Porcentaje
0.135	0.3532	75.0962
BSC	BSP	RSA
0.2227	0.1441	0.6844

NavarraR

Error entrenamiento	BSS	Porcentaje
0.160	0.2772	76.5256
BSC	BSP	RSA
0.2227	0.1610	0.6157

4.1.3-VIENTOS

Primero se realiza la predicción determinista del viento (del tipo Velocidad = 12km/h).

Para el patrón iberiacomp 1.5° con 2 capas ocultas:

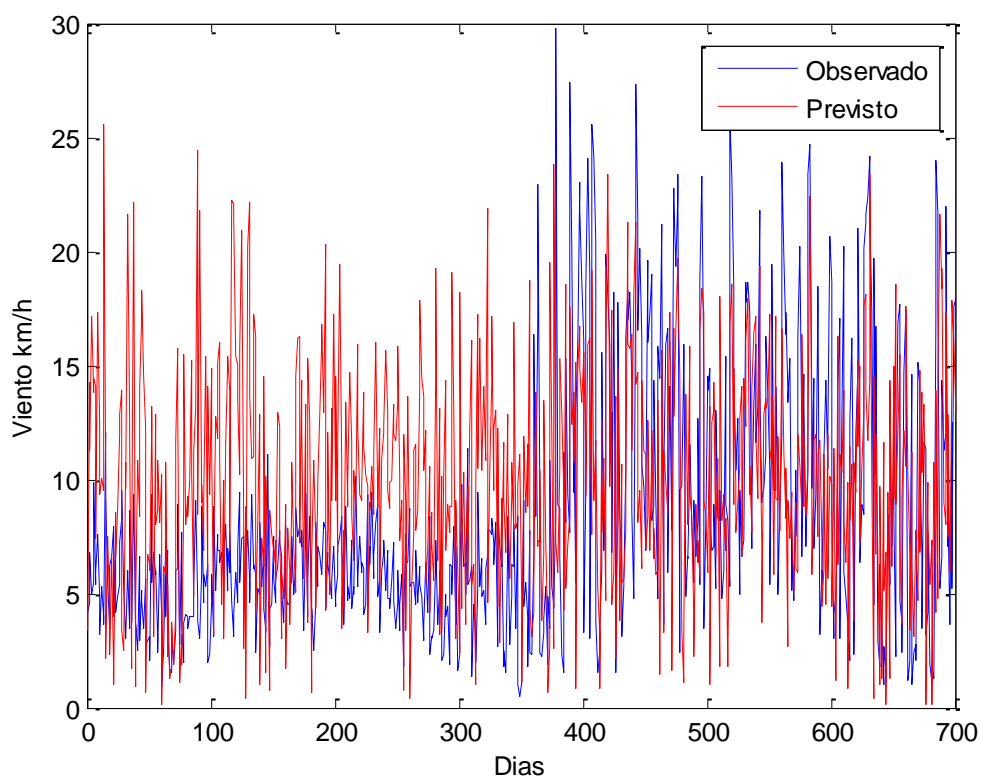


FIG 4.31 OBSERVADOS VS PREVISTOS IBERIACOMP 2 CAPAS

Error entrenamiento	Error test	Error medio
0.0216	1.4923	5.2228

OBSERVADO	PREVISTO		OBSERVADO	PREVISTO
12	15,9516		17,1	9,914
23,9	16,6654		3	11,1264
19,8	11,198		12,6	10,3475
16,4	8,2632		20,2	13,8886
17,3	10,5031		5,9	8,1624
13,9	2,6683		4,9	6,3929
13,4	9,105		3,6	9,8729
15,3	7,4068		3,3	0,7905
5,1	9,4213		2,1	11,4763
7,3	8,4169		12,5	8,1614
4,7	5,1098		16,2	10,6762

10,4	8,96	7	3,7735
7,5	6,0877	2,3	10,7847
10,5	5,9041	12,1	9,4293
16,2	11,9779	14,7	15,2132
20,2	10,6759	21	11,4094
12	8,6883	11,1	14,9432
6,6	16,3331	6,4	7,2603
14,6	8,8269	8,9	10,6918
13,6	11,9485	8,5	12,9041
7,1	8,4383	20,1	17,629
10,2	10,5686	21,3	18,0939
23,4	11,3719	21,6	13,9207
24,7	22,381	22,4	11,7433
13,2	11,6816	24,2	14,5231
9,6	5,7938	21,4	23,4102
14,4	9,9645	17,3	9,8975
7	11,7504	4,5	0,4055
9,3	11,9762	19,7	14,498
17,5	7,5096	10,2	6,7328
18,5	11,1338	16,7	11,9719
3,2	11,6762	2,9	6,7666
6	7,3619	1,8	0,9838
5,4	4,4285	6,3	9,666
10,5	5,9869	3,4	1,2563
10,9	11,0746	1	11,5914
14,3	4,5378	2,7	5,099
10,5	11,9724	1,6	6,7686
12,1	5,6402	1,1	0,1741
20,7	10,1368	8,5	9,3892
18,6	9,7875	5,4	1,2861
8,6	4,3861	13,9	14,3094
3	10,0452	7,1	6,1491
4,1	11,3536	11,5	14,9994
5	1,1559	8,2	9,1561
11	16,2529	2,2	18,5264
14,1	6,2508	8,5	8,9392

TABLA VALORES OBSERVADOS VS PREVISTOS (ALEATORIA)

Para la misma red con entrenamiento de levenberg-marquardt:

Error entrenamiento	Error test	Error medio
0.0051	2.1200	6.1371

Utilizando los mismos patrones con una red de dos capas ocultas con 20 y 10 neuronas en las capas:

Traingdx

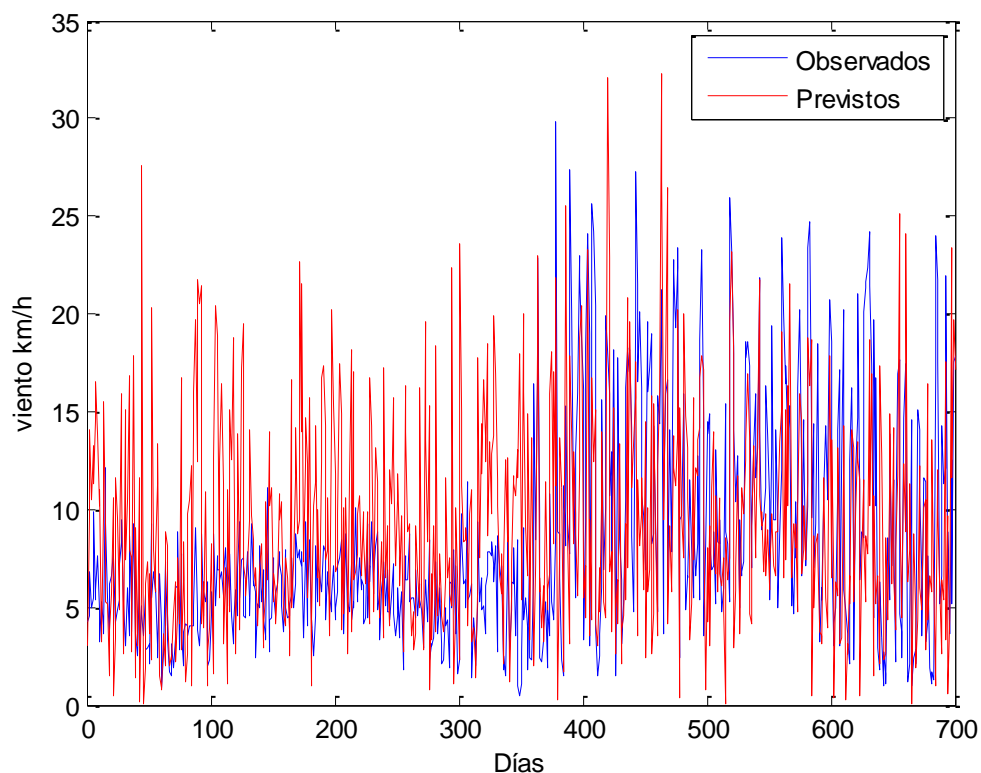


FIG 4.32 OBSERVADOS VS PREVISTOS IBERIACOMP LEVENBERG-MARQUARDT

Error entrenamiento	Error test	Error medio
0.0648	1.4320	4.8805

Levenberg-Marquardt

Error entrenamiento	Error test	Error medio
0.5499	1.2564	4.9262

Probando el patrón de la iberia con componente temporal con la misma estructura de red:

Traingdx

Error entrenamiento	Error test	Error medio
0.0554	0.9927	3.8990

Levenberg-Marquardt

Error entrenamiento	Error test	Error medio
0.0400	1.2512	4.4844

Si se prueba unos patrones más específicos con la misma red:

Patrón Iberia U y V a 1000hPa

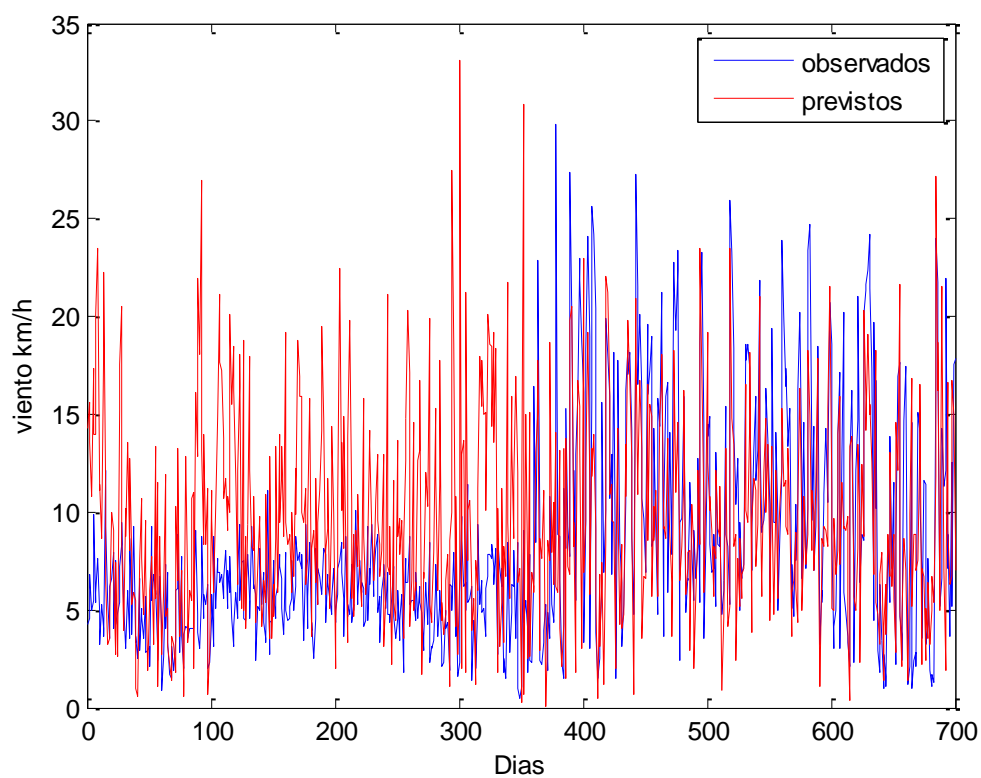


FIG 4.33 OBSERVADOS VS PREVISTOS IBERIAUV

Traingdx

Error entrenamiento	Error test	Error medio
0.0604	1.2568	4.4794

Levenberg-Marquardt

Error entrenamiento	Error test	Error medio
0.2272	1.2680	4.7177

Para un patrón más específico como navarraUV

Traingdx:

Error entrenamiento	Error test	Error medio
0.2709	1.1047	4.3325

Levenberg-Marquardt:

Error entrenamiento	Error test	Error medio
0.1601	1.5418	4.8908

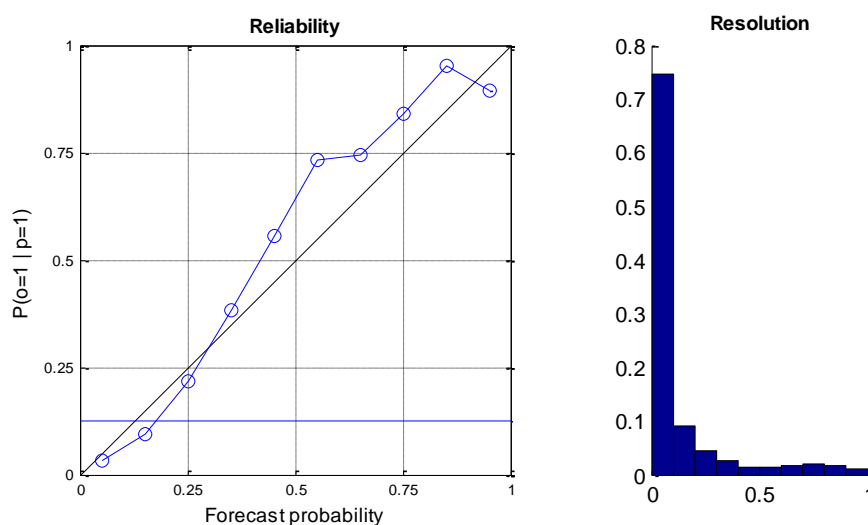
OBSERVADO	PREVISTO		OBSERVADO	PREVISTO
18,6	15,6697		5,4	4,7823
8,6	8,7786		13,9	23,9726
3	4,0625		7,1	21,5427
4,1	26,0957		11,5	33,1916
5	7,9329		8,2	3,6784
11	9,7557		2,2	0,0822
14,1	12,9094		8,5	7,3868
17,1	14,7685		16,9	19,6452
3	10,3476		17,7	10,8376
12,6	10,438		4,7	14,0644
20,2	10,975		3,7	1,9959
5,9	3,0343		2,4	6,3522
4,9	5,2518		14,7	5,5211
3,6	10,5629		17,4	17,8379
3,3	4,4506		16,9	17,7721
2,1	5,7741		1,2	7,0621
12,5	5,2357		2,1	2,2936
16,2	9,141		4,2	7,9239
7	3,9314		14,6	11,0574
2,3	4,159		1	6,3318
12,1	14,5629		2,4	11,7821
14,7	9,1405		2,8	6,7656
21	7,528		2,1	5,1046
11,1	11,6147		15,1	5,6136

6,4	1,2952	14,1	12,6269
8,9	4,614	10,7	6,5967
8,5	15,4208	6,8	14,293
20,1	10,7367	3,5	11,3358
21,3	20,6282	11,6	4,1149
21,6	11,4081	11,4	5,9202
22,4	18,6501	3,2	7,6354
24,2	17,1746	4,2	14,8546
21,4	29,5879	7,6	3,9992
17,3	12,081	1,9	4,7965
4,5	1,2472	1,1	6,6546
19,7	12,2735	1,7	2,8634
10,2	9,6466	1,3	7,1882
16,7	11,8522	17	8,6931
2,9	1,0412	24	15,5943
1,8	6,24	21,7	22,5994
6,3	7,2759	17,6	13,8162
3,4	7,1371	5,2	18,1267
1	5,5119	6,6	4,6409
2,7	4,9713	14,3	14,5025

Como se observa los resultados son algo mejores que en el caso de la predicción determinista de la precipitación. También como en ese caso se va a probar a hacer una predicción probabilística, poniendo un umbral de separación.

En este caso se prueban para diferentes patrones la separación en dos clases vientos suaves-moderados y moderados a fuertes. Con una red similar a las anteriores es decir con dos capas ocultas, ya que con 1 capa oculta da peores resultados en las pruebas realizadas.

Para un patrón iberia completo de 1,5°:



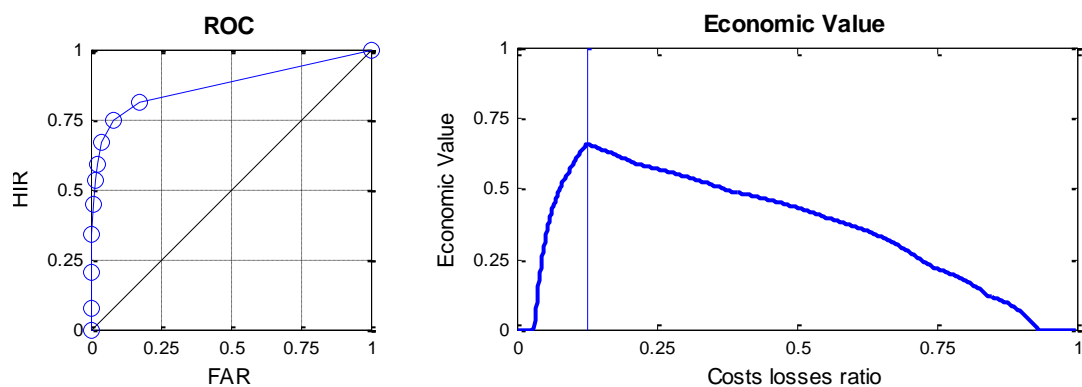
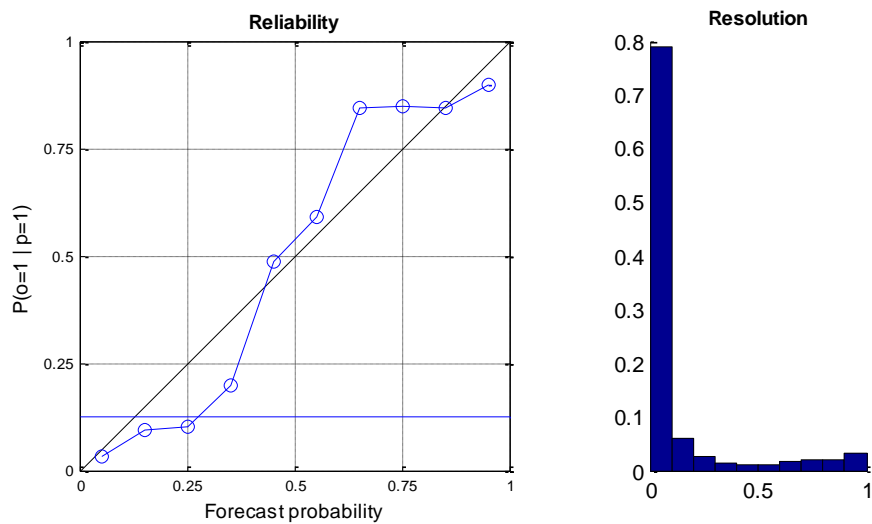


FIG 4.34 GRÁFICOS VALIDACIÓN IBERIACOMP 2 CAPAS

Error entrenamiento	BSS	Porcentaje
0.0362	0.4647	83.8074
BSC	BSP	RSA
0.1099	0.0588	0.7388

Para el patrón iberiatemp:



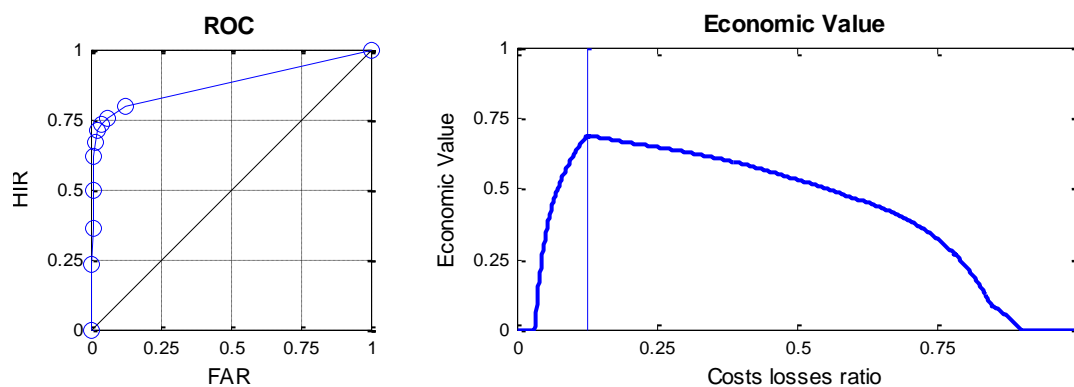
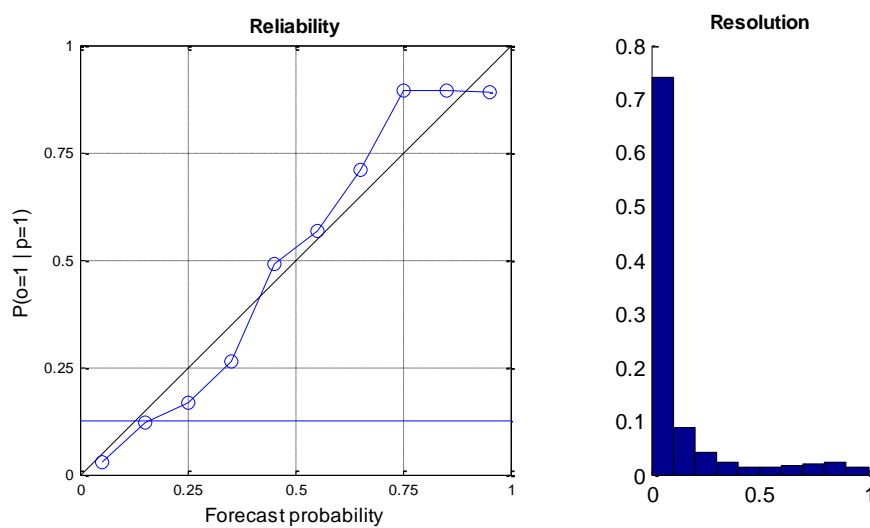


FIG 4.35 GRÁFICOS VALIDACIÓN IBERIATEMP 2 CAPAS

Error entrenamiento	BSS	Porcentaje
0.0362	0.5301	85.0109
BSC	BSP	RSA
0.1099	0.0516	0.7467

Para un patrón más específico como el de iberia 1000 con las componentes U y V:



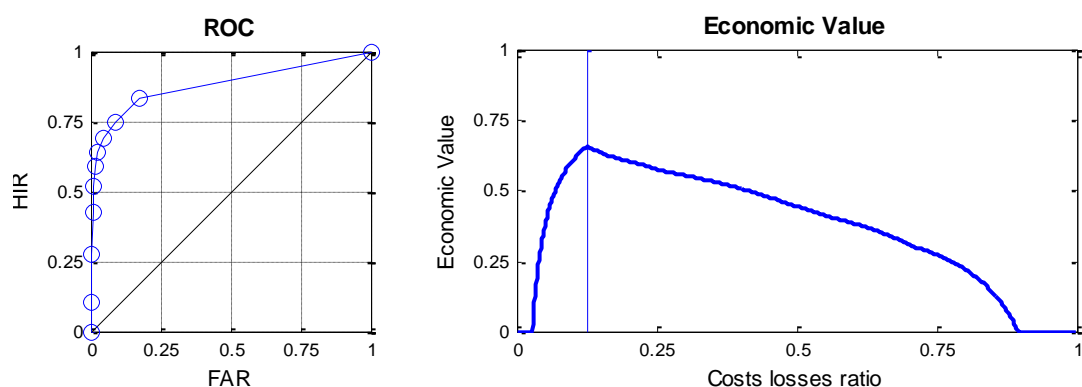


FIG 4.36 GRÁFICOS VALIDACIÓN IBERIA1000UV

Error entrenamiento	BSS	Porcentaje
0.0362	0.4805	85.0109
BSC	BSP	RSA
0.1099	0.0571	0.7609

Se observa que se obtienen unos resultados aceptables, con un porcentaje de acierto superior al 85%.

4.1.4-DIRECCIÓN DEL VIENTO.

Por otra parte también se puede clasificar la dirección del viento, en este caso como ya se ha explicado en un apartado anterior se va a clasificar los vientos en 4 clases distintas. La red neuronal en este caso se plantea de una forma distinta a las utilizadas con anterioridad. Se asigna a las salidas un vector con el índice a 1 en el lugar que corresponde a la clase.

Así se elige el índice que tiene valor máximo al simular.

Clase original	Salida (máximo a 1)	Clase prevista
4	[0 0 0 1 0]	4
4	[0 0 0 1 0]	4
4	[0 0 0 1 0]	4
2	[0 0 0 1 0]	4
1	[1 0 0 0 0]	1
1	[1 0 0 0 0]	1
2	[0 1 0 0 0]	2
3	[0 1 0 0 0]	2

4	[0 0 0 1 0]	4
4	[0 0 0 1 0]	4
4	[0 0 0 1 0]	4
2	[0 1 0 0 0]	2
3	[0 1 0 0 0]	2
1	[1 0 0 0 0]	1
2	[0 1 0 0 0]	2
4	[0 0 0 1 0]	4

TABLA- EJEMPLO CLASIFICADOR

Para el patrón de Iberia con componentes temporales que se muestra en la Figura 4.37 se obtiene un acierto del 88.8144%.

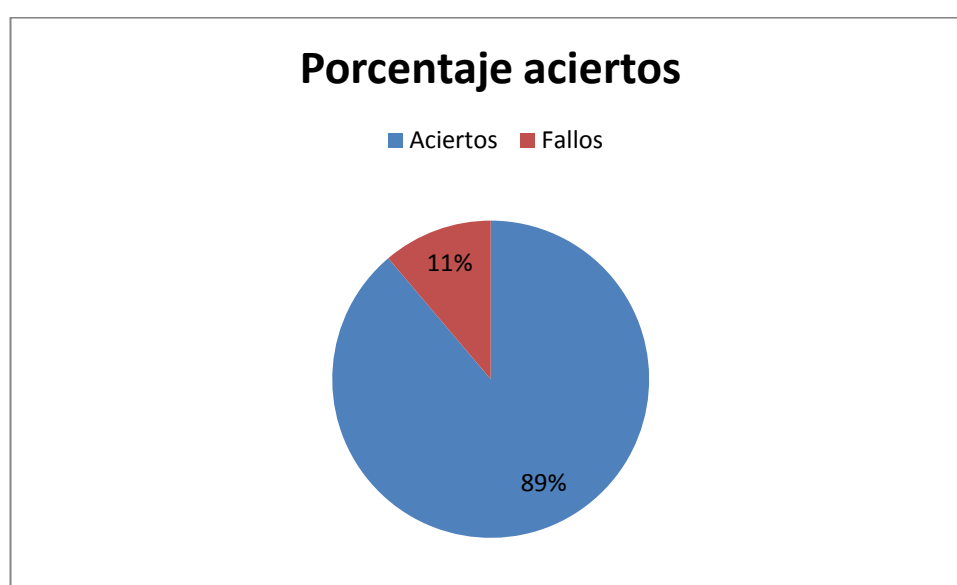


FIG 4.37- ACIERTOS PATRÓN IBERIATEMP

Para el patrón de Iberia con componentes temporales pero considerando solamente U y V se obtiene un acierto del 89.0614% y se muestra en la Figura 4.38.

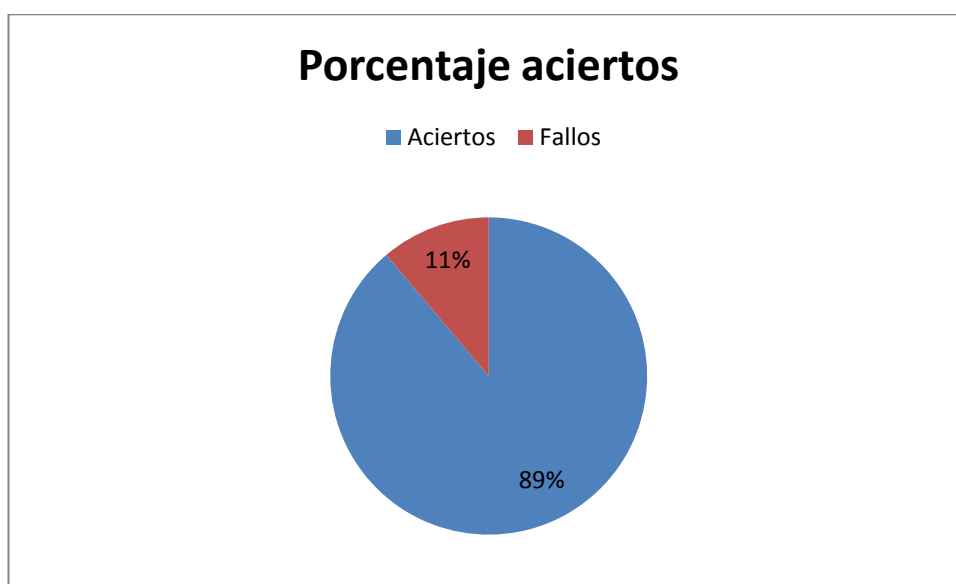


FIG 4.38- ACIERTOS PATRÓN IBERIA1000UV

Para un patrón más específico como navarraUV850 el acierto es 62.3500% como se muestra en la Figura 4.39.

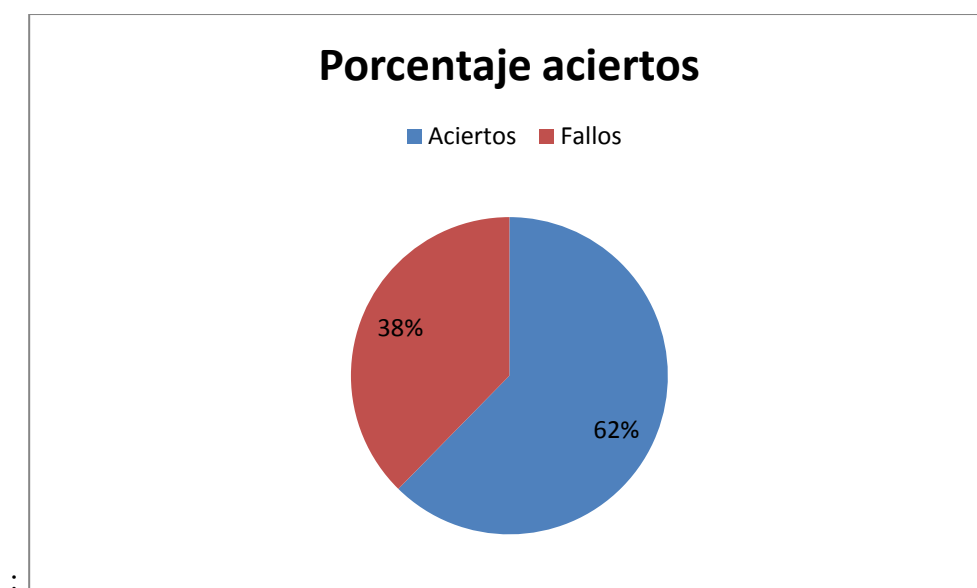


FIG 4.39- ACIERTOS PATRÓN NAVARRAU850

Y por último para uno más genérico el de la NAO se obtiene un acierto del 76.9584 % como se muestra en la Figura 4.40.

:

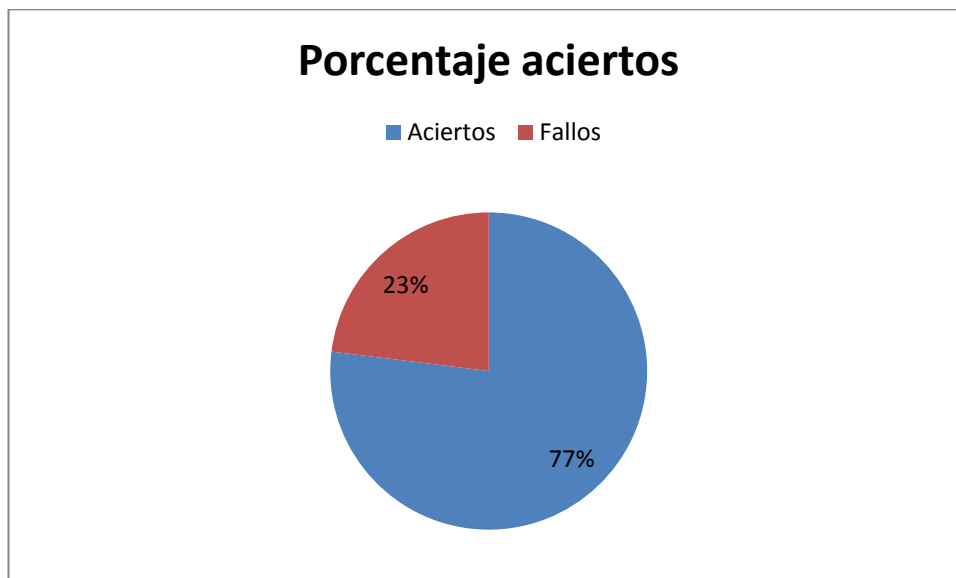


FIG 4.40- ACIERTOS PATRÓN NAO

4.2-REGRESIÓN

4.2.1-TEMPERATURA

Temperatura máxima con patrón iberia 1°:

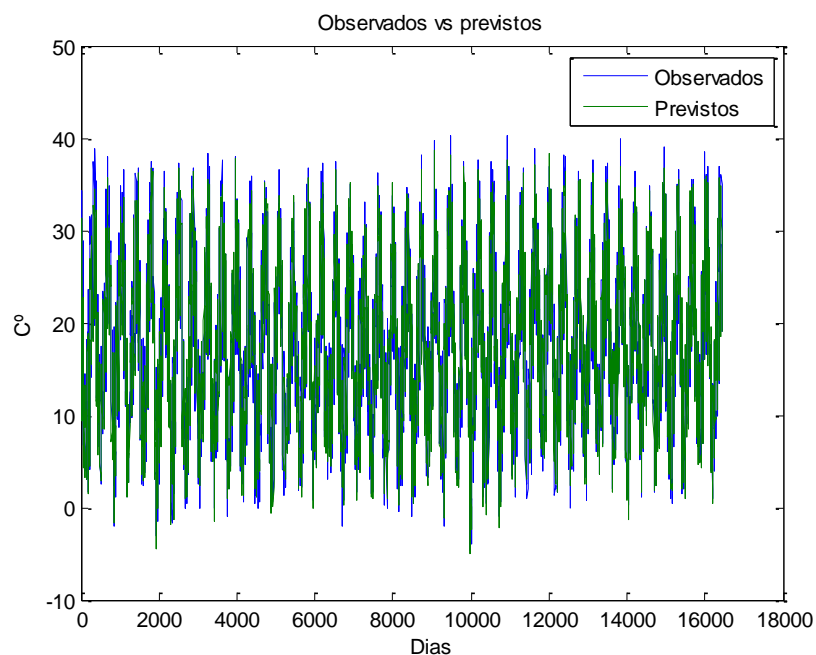


FIG 4.41- REGRESIÓN IBERIA

Error cuadrático medio	Error medio
4.4666	1.6324

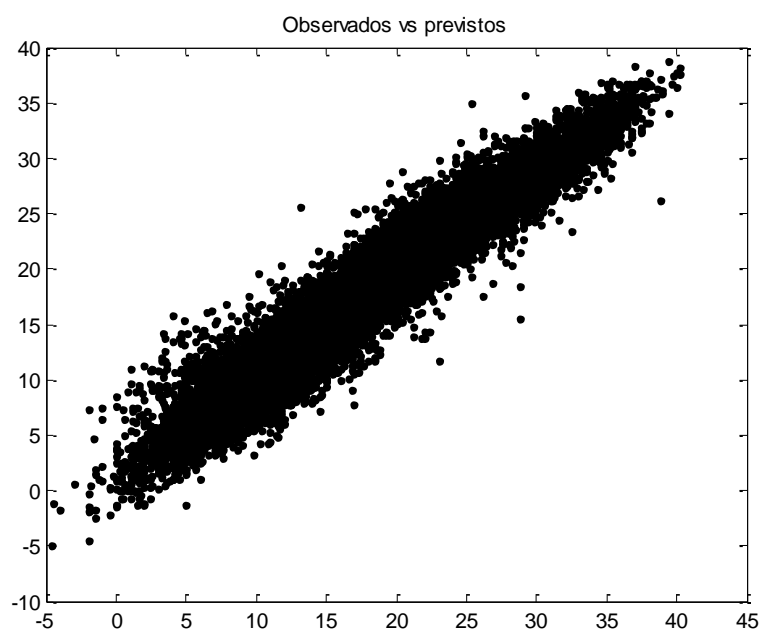


FIG 4.42- REGRESIÓN IBERIA

Para el patrón Iberiatemp:

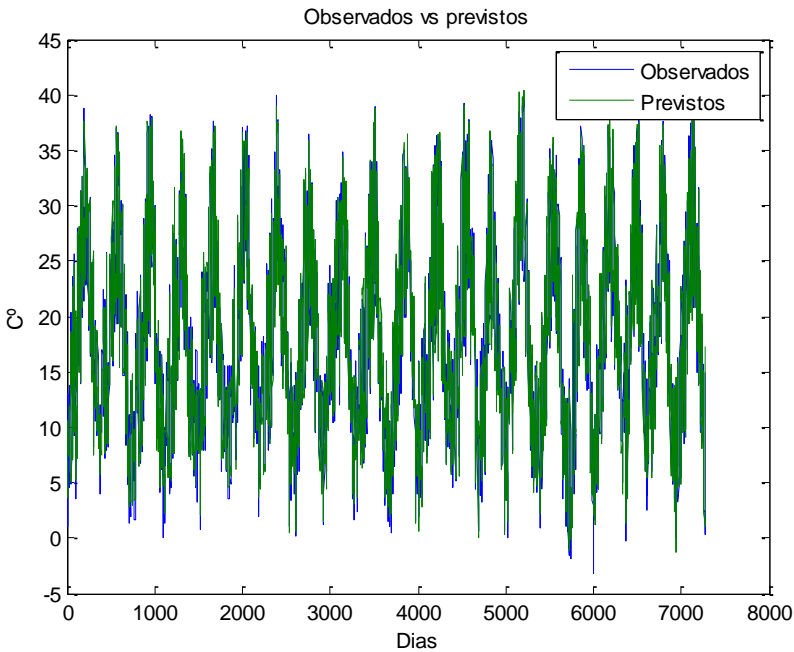


FIG 4.42- REGRESIÓN IBERIATEMP

Error cuadrático medio	Error medio
1.7757	1.0342

Para el patron Iberiacomp:

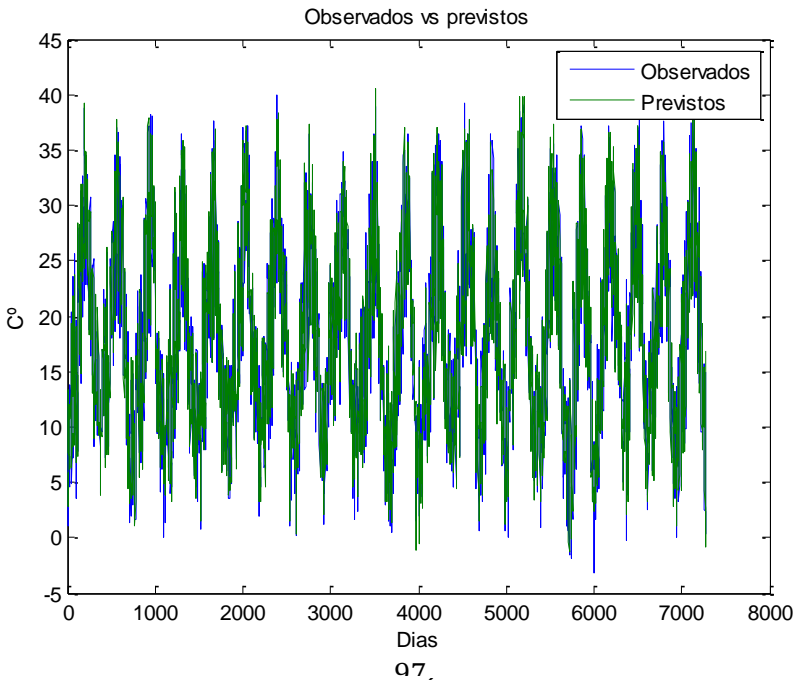


FIG 4.43- REGRESIÓN IBERIACOMP

Error cuadrático medio	Error medio
3.7614	1.5322

Para el patrón de NavarraT:

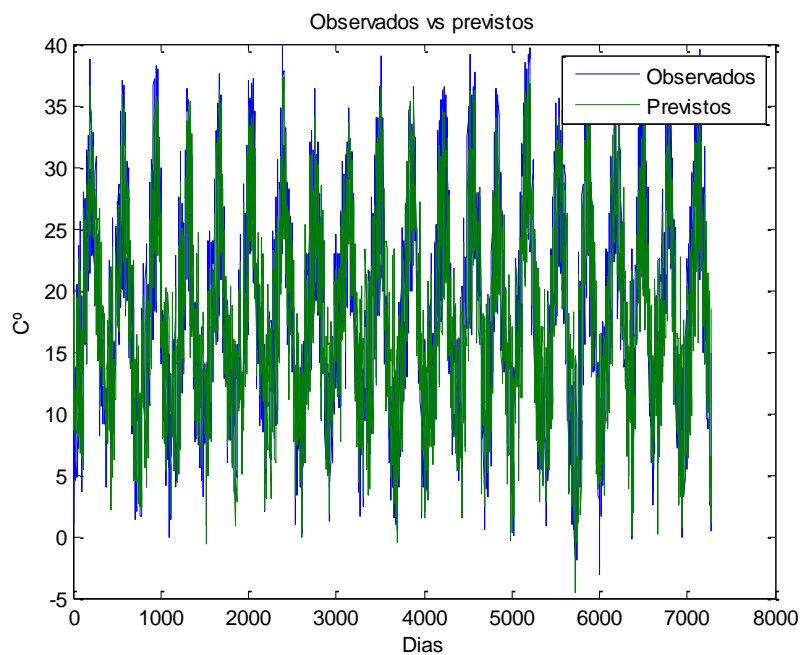


FIG 4.44- REGRESIÓN NAVARRAT

Error cuadrático medio	Error medio
31.6720	4.5237

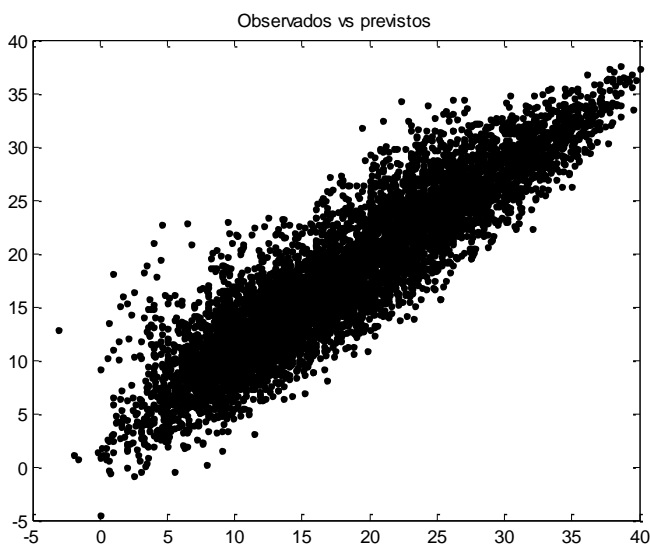


FIG 4.45- REGRESIÓN NAVARRAT

Temperatura mínima con patrón iberia 1°:

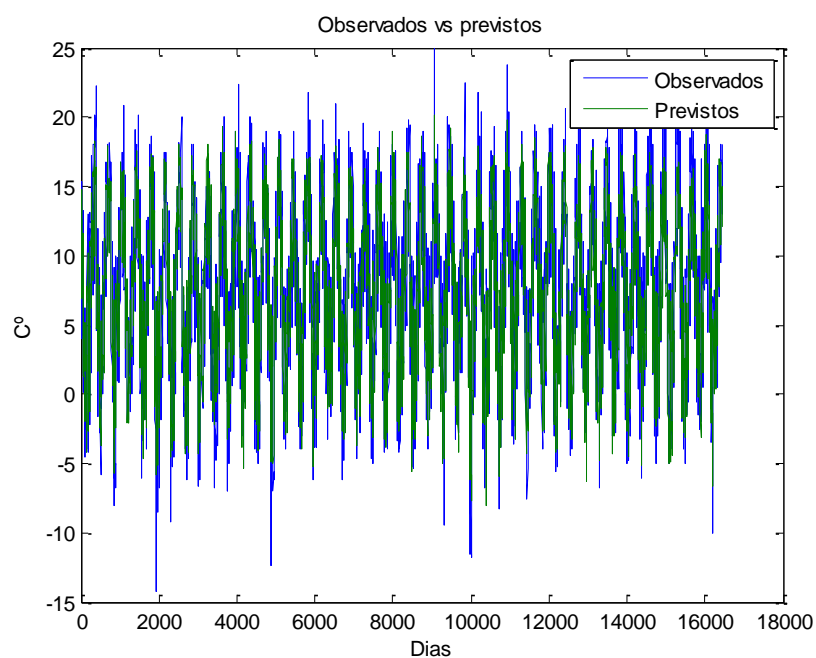


FIG 4.46- REGRESIÓN IBERIA (MÍNIMA)

Error cuadrático medio	Error medio
4.2581	1.6237

Para el patrón Iberiatemp:

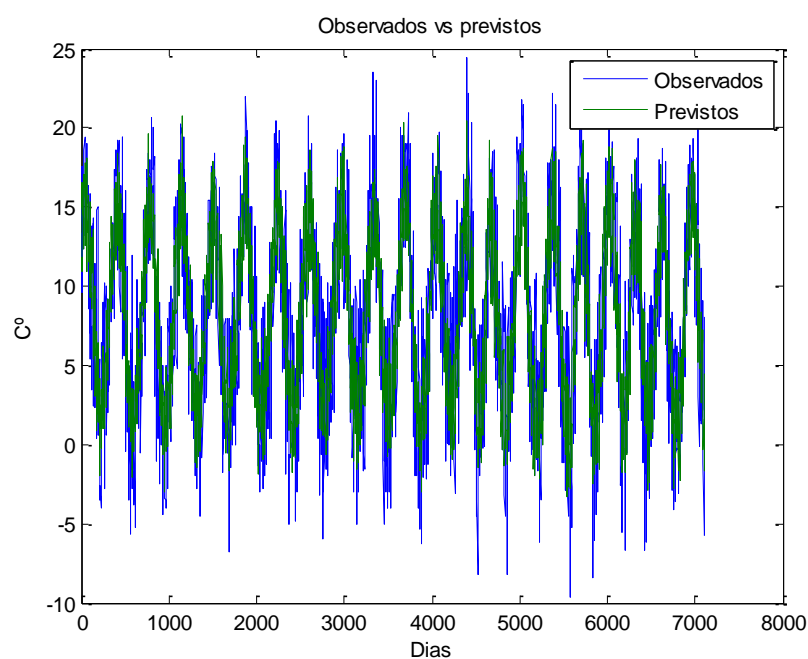


FIG 4.47- REGRESIÓN IBERIATEMP (MÍNIMA)

Error cuadrático medio	Error medio
8.0319	2.2679

Para ibericomp:

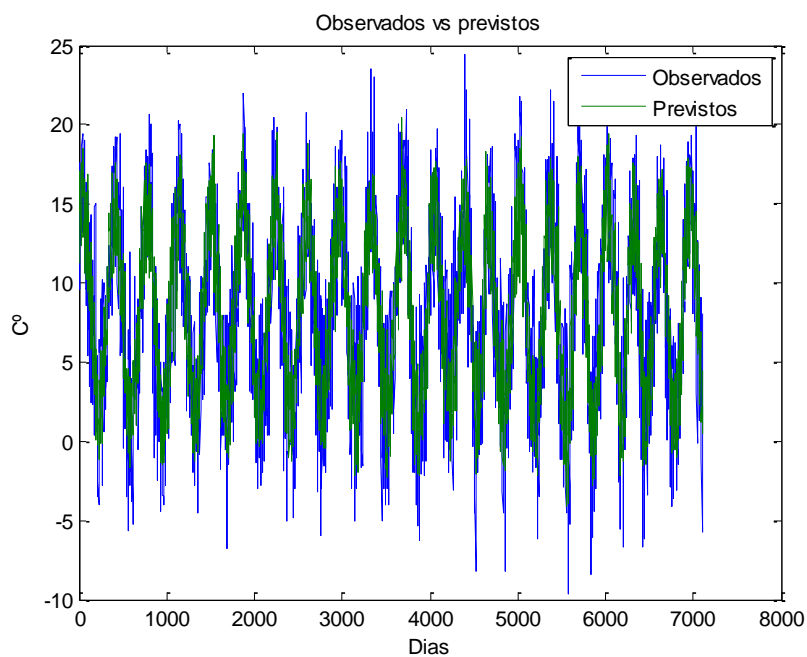


FIG 4.48- REGRESIÓN IBERIACOMP (MÍNIMA)

Error cuadrático medio	Error medio
8.3409	2.3117

Para NavarraT:

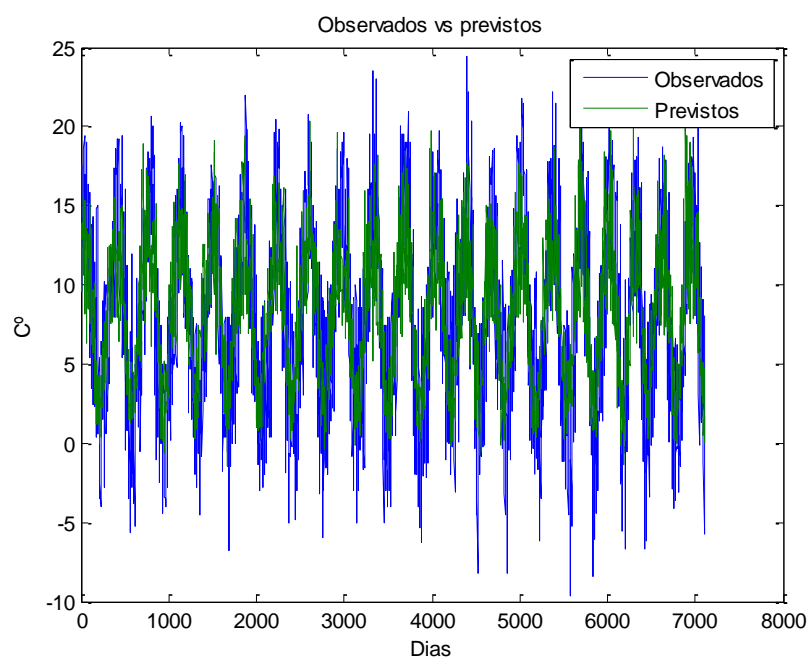


FIG 4.49- REGRESIÓN NAVARRAT (MÍNIMA)

Error cuadrático medio	Error medio
16.9594	3.3110

4.2.2-PRECIPITACIÓN

Para iberiategp:

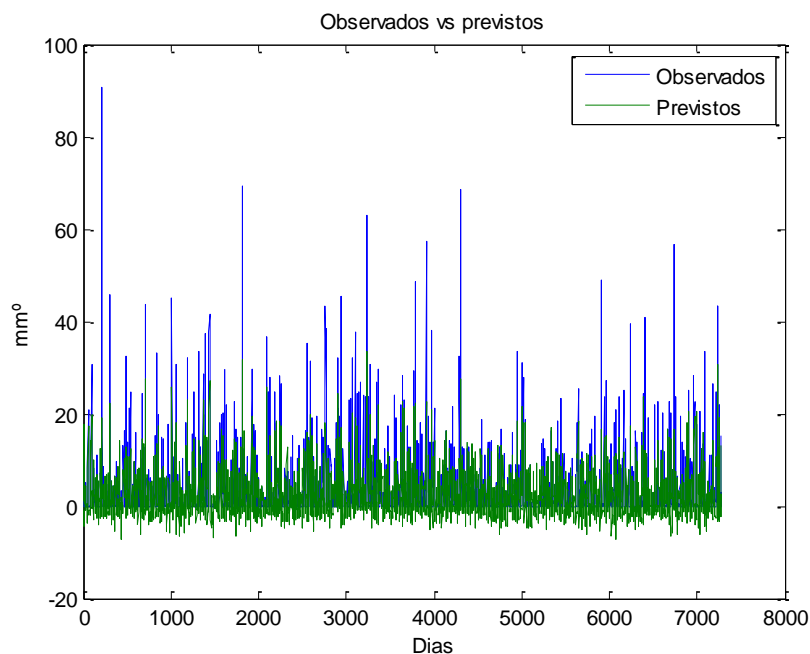


FIG 4.50- REGRESIÓN IBERIATEMP (PRECIPITACIÓN)

Error cuadrático medio	Error medio
12.0040	2.1737

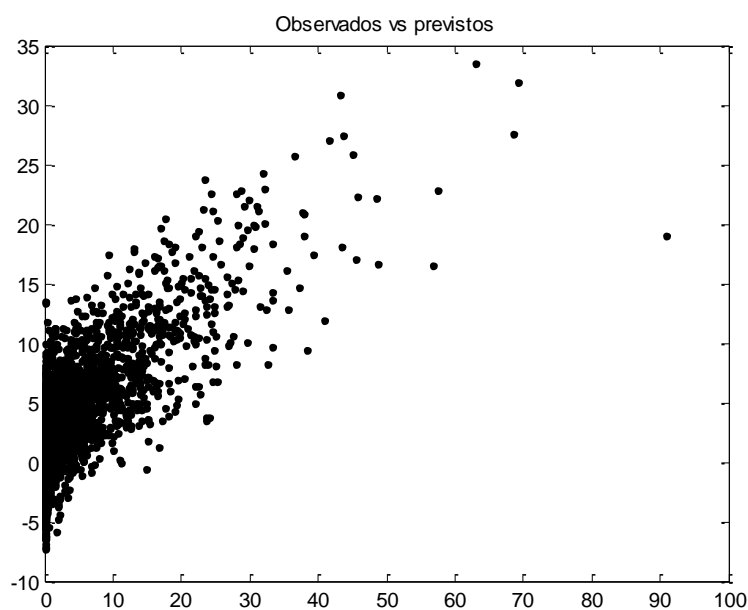


FIG 4.51- REGRESIÓN IBERIATEMP (PRECIPITACIÓN)

Error cuadrático medio	Error medio
12.0040	2.1737

Para Iberiacomp:

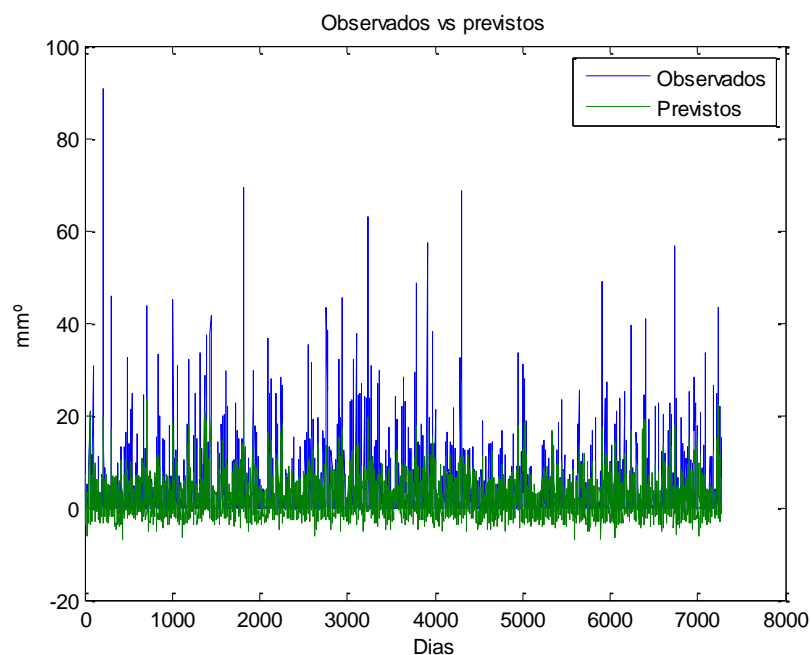


FIG 4.52- REGRESIÓN IBERIACOMP (PRECIPITACIÓN)

Error cuadrático medio	Error medio
15.5164	2.4793

Para el patrón navarraR:

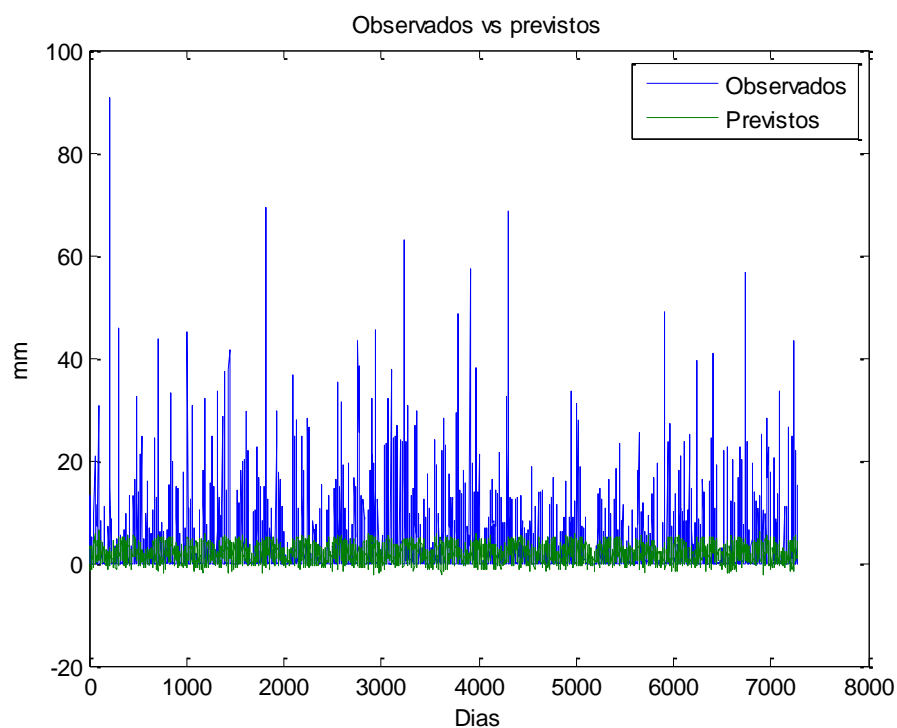


FIG 4.53- REGRESIÓN NAVARRAR (PRECIPITACIÓN)

Error cuadrático medio	Error medio
23.9123	2.5125

5.2.3-VIENTO

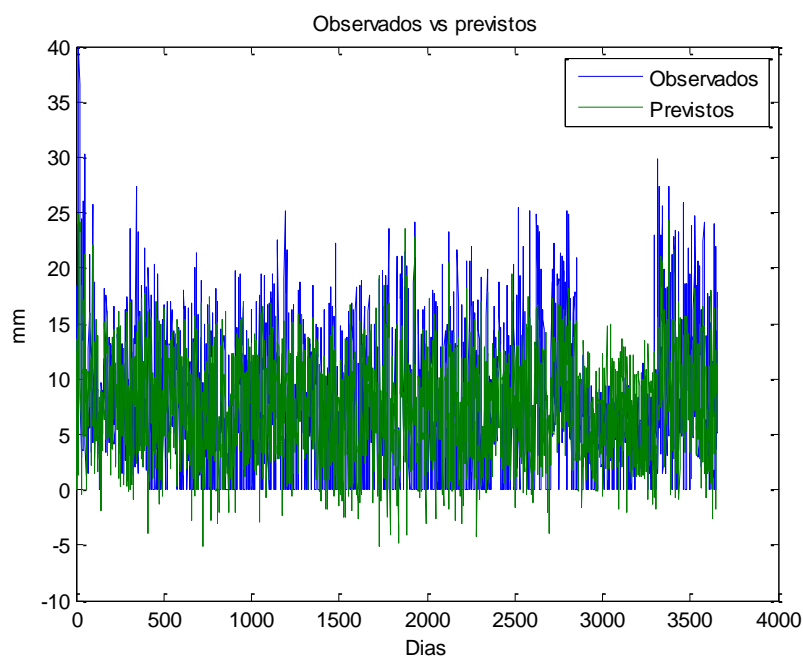


FIG 4.54- REGRESIÓN IBERIA (VIENTO)

Error cuadrático medio	Error medio
21.4589	3.6959

Para el patrón iberiategp:

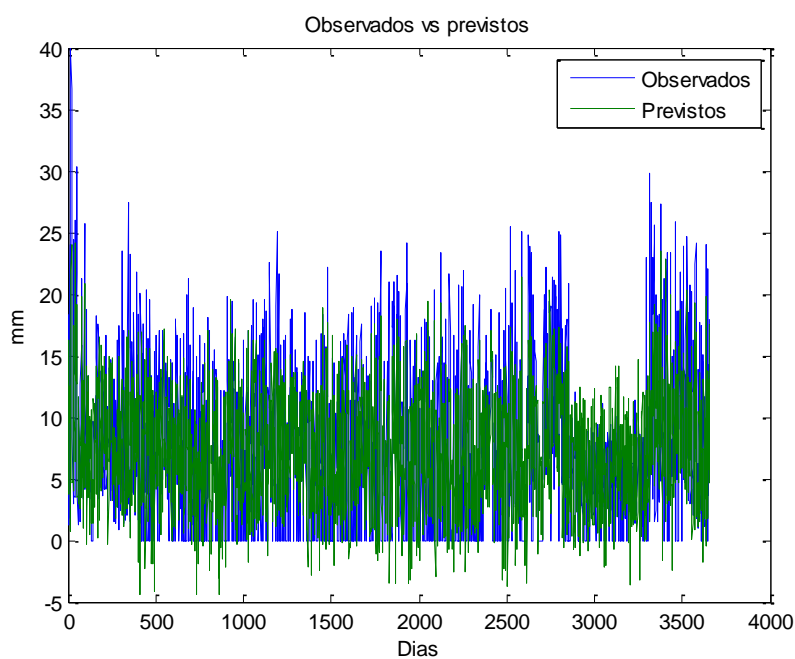
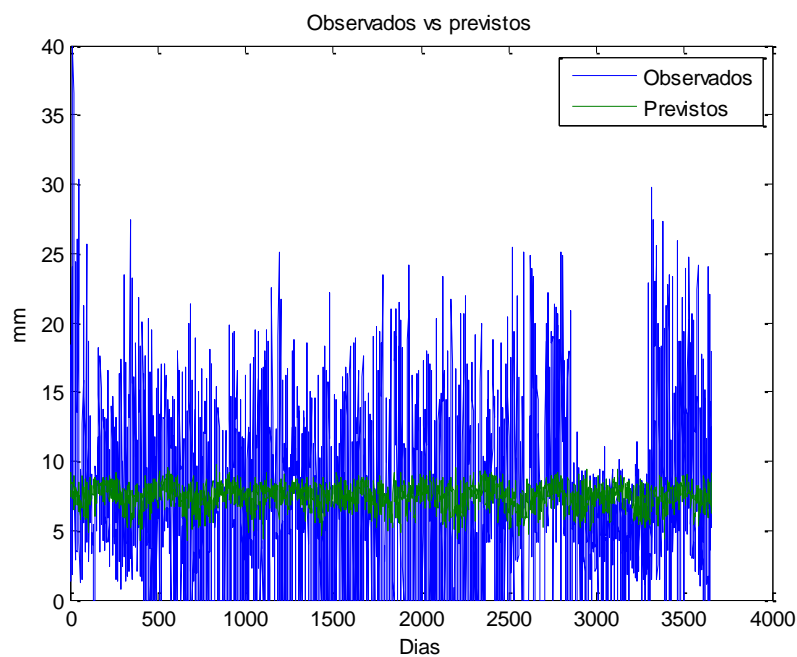


FIG 4.55- REGRESIÓN IBERIATEMP (VIENTO)

Error cuadrático medio	Error medio
20.9422	3.6622

Para navarraUV850:



5.56- REGRESIÓN NAVARRAUV850 (VIENTO)

Error cuadrático medio	Error medio
37.7806	4.9898

5-Herramienta

Una vez que se han realizado las distintas pruebas con los patrones y datos de las estaciones, se va a proceder a integrar todo lo utilizado en una herramienta con interfaz gráfico para hacer más sencilla su utilización. Se van a detallar las diferentes opciones de la herramienta y se especificarán algunos aspectos de cómo funcionan los *scripts* más importantes.

Esta herramienta se divide en tres partes principales: la primera es sobre observación en tiempo real de los datos de la estación para la que se ha trabajado, es decir la de Pamplona. Por otro lado se tiene un apartado de observación de datos históricos, tanto datos propios de la estación como patrones atmosféricos a diferentes niveles de presión.

Por último también se tiene una herramienta para integrar las predicciones utilizando las herramientas generadas en las pruebas.



FIG 5.1 MENÚ PRINCIPAL

En la Figura 5.1 se pueden observar en la pantalla de inicio las tres opciones que se han descrito anteriormente. A continuación se van a describir las funcionalidades de cada uno de los apartados accesibles desde el menú principal.

5.1-OBSERVACIÓN

En este apartado se puede consultar la situación actual de la estación para la que se ha estado realizando todas las pruebas.

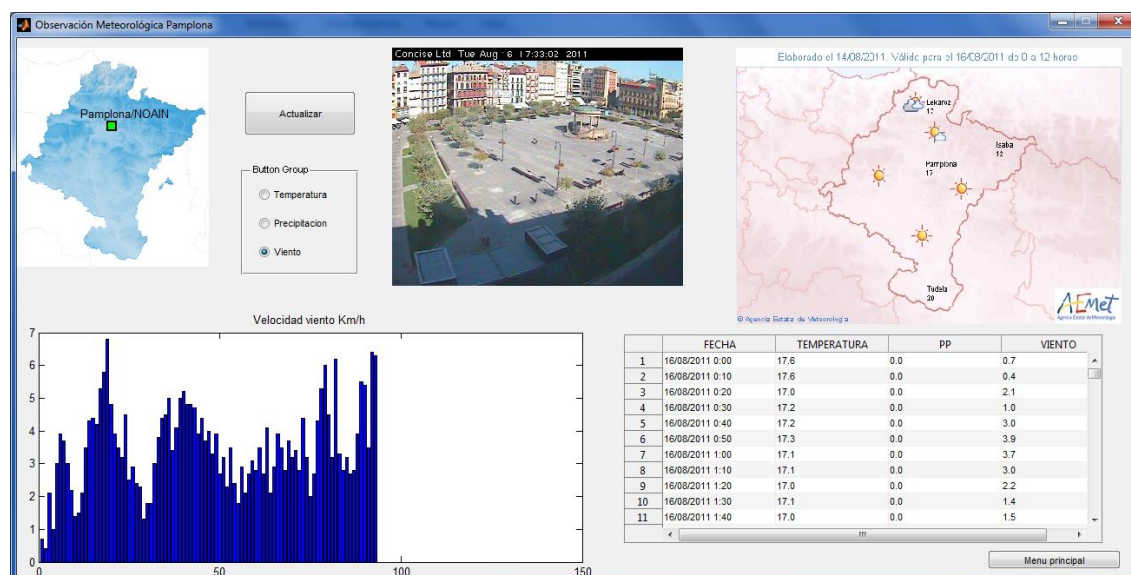


FIG 5.2 OBSERVACIÓN

En la parte de arriba a la izquierda hay un mapa en el que se indica la situación de la estación para la que se observan los datos, por ejemplo si se extendiese el estudio a otras estaciones sería fácil indicarlo en el mapa ya que se trata de un gráfico con coordenadas.

En la parte central y arriba de la ventana se ha incorporado una imagen en tiempo real de una webcam de Pamplona, lo mismo que se ha dicho anteriormente, se podrían obtener imágenes de otras estaciones.

En la parte derecha y arriba la herramienta se incluye un pronóstico de AEMET para Navarra para el día actual de observación.

En la parte de abajo se tiene un gráfico en el que se pueden mostrar los datos del día actual cada 10 minutos, mediante los “radiobutton” de arriba se puede elegir entre la precipitación, la temperatura y la velocidad del viento.

En la parte derecha aparecen los mismos datos que en la gráfica pero tabulados y con la hora también cada 10 minutos.

Por último existe un botón que permite actualizar los datos que se muestran en la pantalla, es decir actualiza la imagen y los datos de la estación. Estos se obtienen de un fichero Excel que se descarga desde meteo-navarra. Una barra de progreso indicará que la actualización está en curso. Mediante un script en MATLAB se pueden descargar los datos desde la URL especificada y leerlos para procesarlos en el gráfico o la tabla. Al descargarse en fichero Excel es necesario tener instalada esta herramienta para poder visualizar los datos.

Desde el botón volver al menú se puede volver al menú principal.

5.2-HISTÓRICOS

En este apartado se pueden visualizar tanto los datos históricos de la estación de estudio como los patrones atmosféricos utilizados.

Para ello cuando se accede desde el menú principal aparece una pantalla como la siguiente que permite elegir lo que se desea observar.

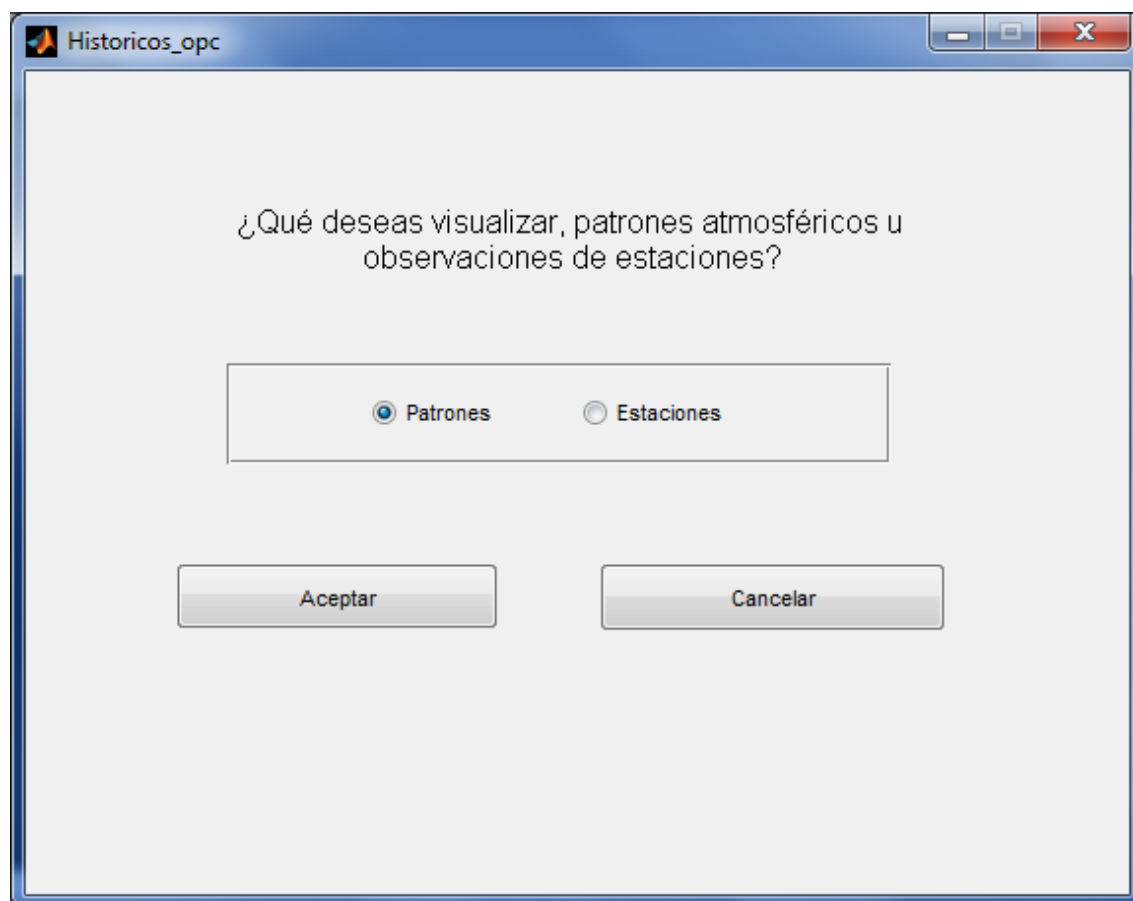


FIG 5.3 OPCIONES HISTÓRICOS

Si se cancela la opción se regresa al menú principal.

5.2.1-ESTACIONES

Si se elige *estaciones* se pasa a una pantalla como la de la Figura 5.4. En la parte derecha se tiene las distintas variables a visualizar, que son temperatura máxima, temperatura mínima, precipitación acumulada, velocidad del viento y dirección del viento.

Una vez seleccionada la variable hay que indicar las fechas entre las que se quiere obtener la información, el formato de entrada de estas es en inglés, Ej: “14-Aug-2000”.

Como no se tienen datos desde las mismas fechas para todas las variables (para el viento no hay datos registrados en la estación antes del año 2000), se advierte con un cuadro de diálogo las fechas que hay que introducir en estos casos concretos. Si falta de introducir la fecha también se avisará de la falta mediante un cuadro de diálogo.

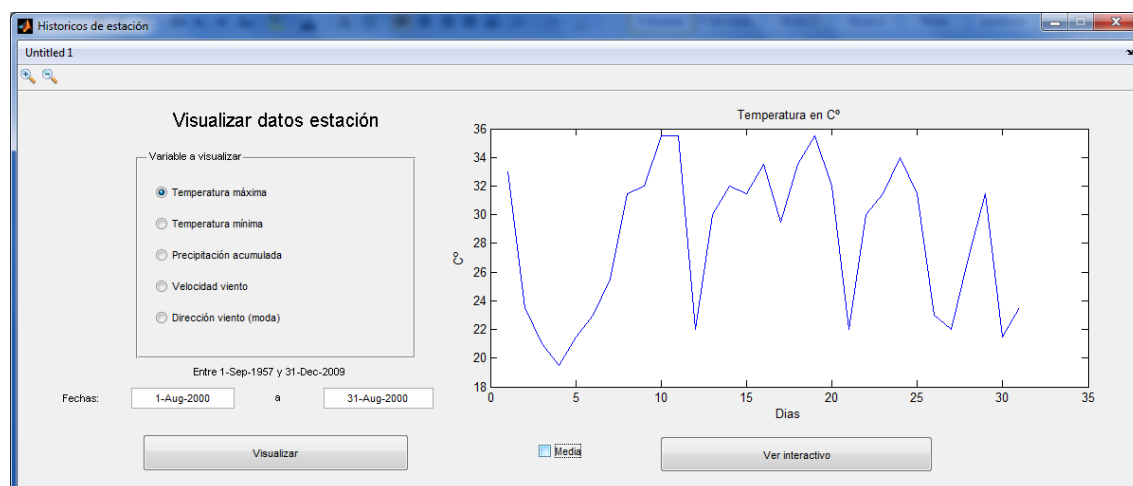


FIG 5.4 VISUALIZACIÓN ESTACIÓN

Una vez seleccionada la variable y las fechas, ya es posible visualizar los datos en la gráfica de la izquierda. Los datos se pueden cambiar rápidamente.

Existe también la opción de visualizar la media de una variable para el periodo en comprendido entre la fechas seleccionadas. Se indica mediante una línea en el gráfico que marca el valor medio de la variable escogida como se ve en la Figura 5.5.

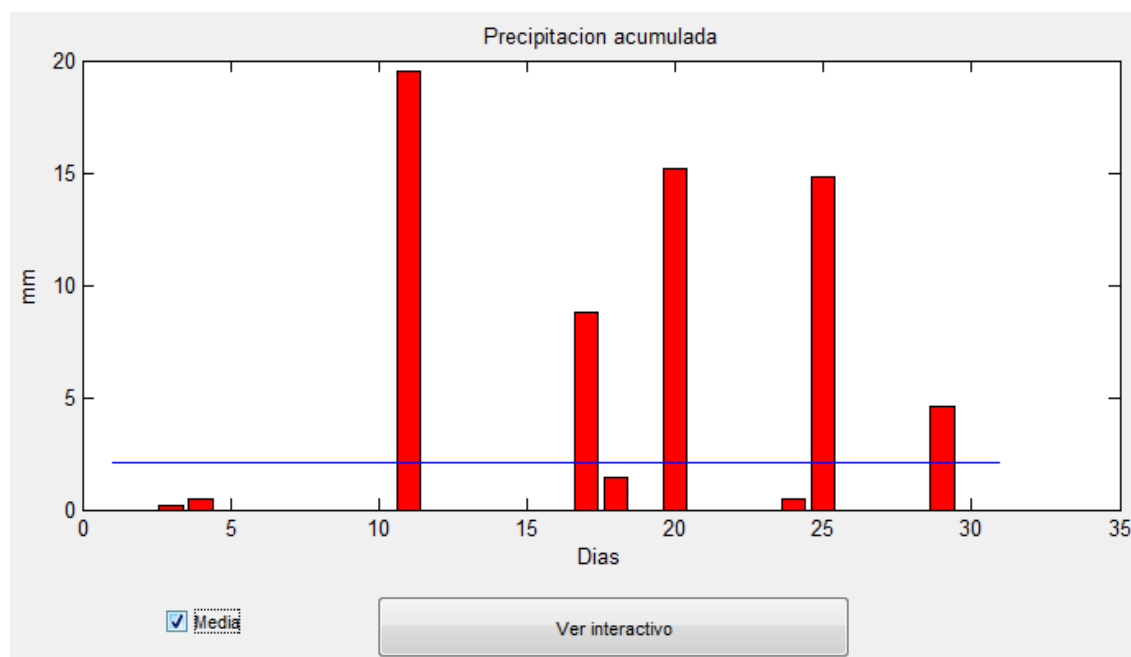


FIG 5.5 DETALLE GRÁFICO ESTACIÓN

Otra opción para visualizar los datos es un gráfico interactivo que permite visualizar los datos avanzando día a día con una imagen de la situación de la estación. El color va variando según el valor de la variable y permite hacer una idea más intuitiva de los valores que va tomando la variable en el periodo seleccionado.

Como se puede ver en la Figura 5.6 mediante los botones con flechas es posible avanzar o retroceder en el periodo, una barra roja nos indica el punto en el gráfico.

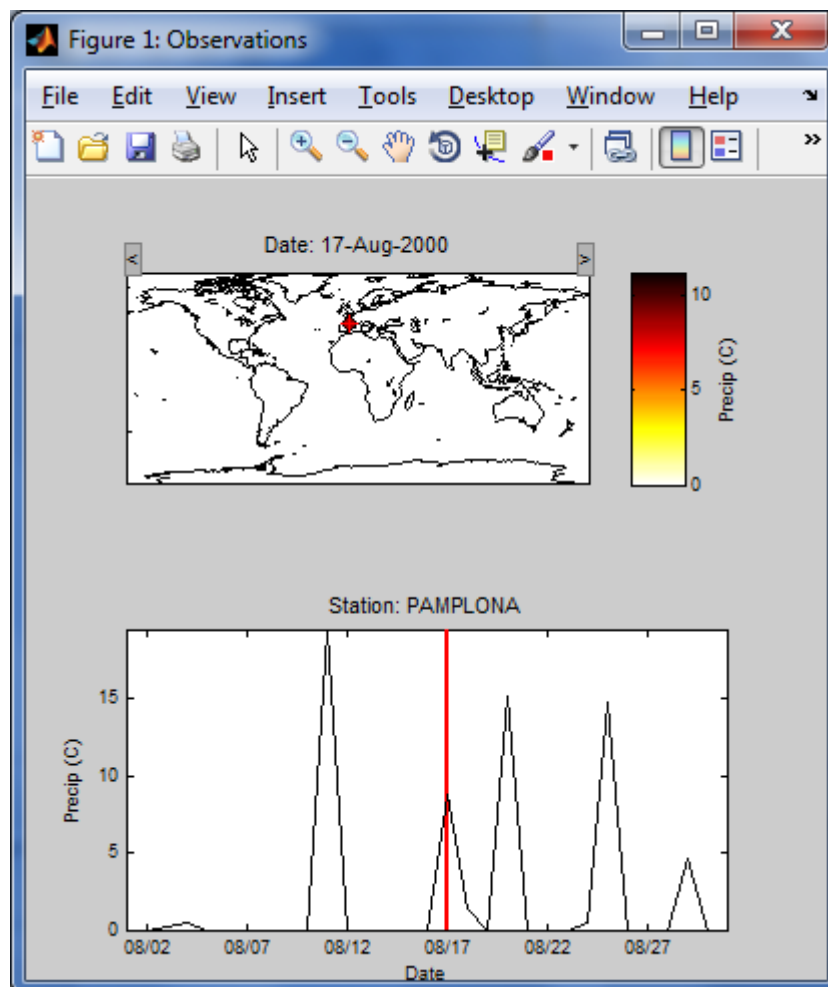


FIG 5.6 ESTACIÓN INTERACTIVA

Si se insertan más estaciones existe la posibilidad de cambiarlas haciendo click con el botón derecho del ratón.

5.2.2-PATRONES

Por otro lado si se eligen los patrones nos aparece una pantalla para realizar una elección de la rejilla a visualizar, las opciones que hay son las que se han utilizado en los experimentos.

Como se puede apreciar en la Figura 5.7 se seleccionan los patrones mediante “radiobutton”. También se puede retroceder al menú principal.

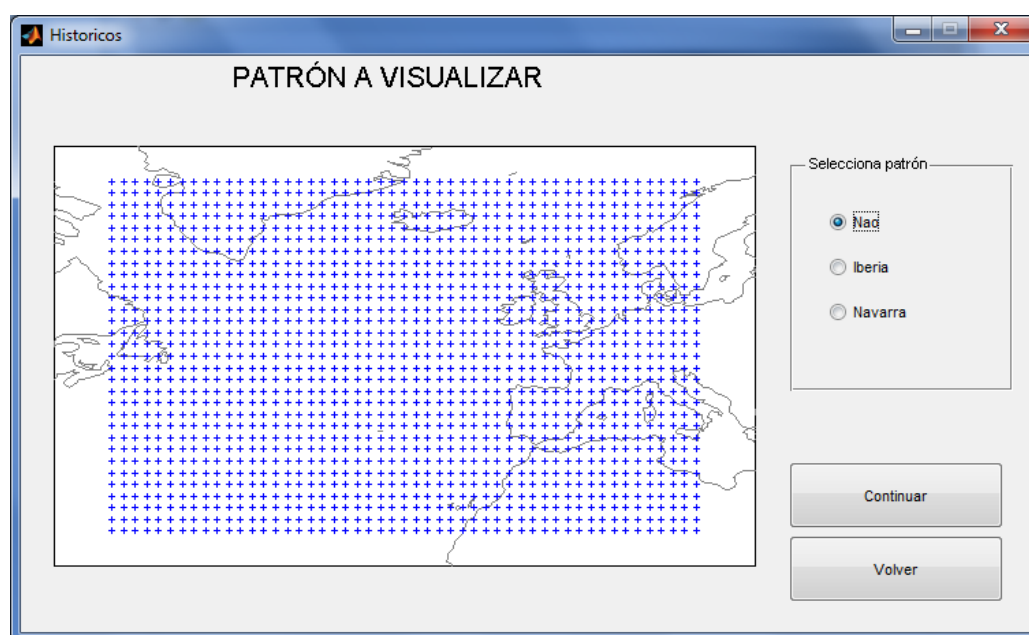


FIG 5.7 SELECCIÓN DE PATRÓN

Una vez se selecciona el patrón que se quiere observar, pulsando el botón de continuar se accede a una pantalla para seleccionar las variables que se quieren visualizar.

The screenshot shows a software window titled "naosel" with a subtitle "Variables a visualizar". It features two main sections for selection. The first section, labeled "Altura", contains two radio buttons: "850 hPa" (which is selected) and "500hPa". The second section, labeled "Variable", contains five radio buttons: "Geopotencial", "Temperatura", "Humedad relativa", "Velocidad u viento" (which is selected), and "Velocidad v viento". Below these sections, there is a text label "Fechas desde 1-Jan-1989 hasta 31-Dec-2009" and two empty input fields labeled "Desde" and "hasta". At the bottom of the window are two buttons: "Continuar" and "Volver".

FIG 5.8 SELECCIÓN DE VARIABLES A VISUALIZAR

Se seleccionara la altura, la variable y la fecha en las que se quiere visualizar el mapa de altura.

Según del modelo seleccionado se podrán visualizar unas variables a una altura u otra dependiendo de la disponibilidad en los ficheros GRIB utilizados durante las pruebas.

Los mapas tiene la siguiente forma como se muestra en la Figura 5.8:

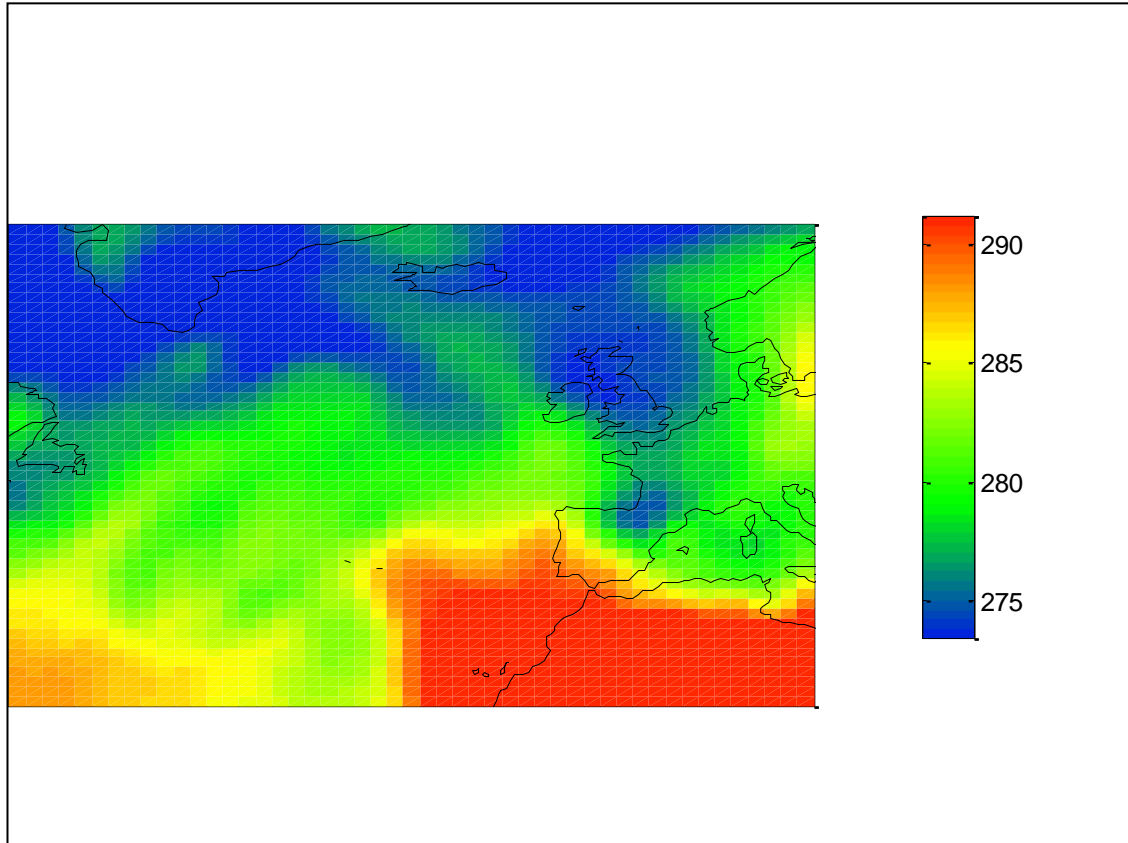


FIG 5.9 MAPAS EN ALTURA DE PATRONES

5.3-PREDICCIÓN

En este apartado se realiza la predicción para tres días, utilizando las redes neuronales entrenadas en las pruebas realizadas. Para realizar esto es necesario generar patrones similares a los utilizados en el entrenamiento de las redes neuronales, para automatizar este proceso se han utilizado dos *scripts* en MATLAB.

```
load 'actualizacion.dat';
data=num2str(actualizacion);
fecha=date;
fecha=datestr(fecha,'yyymmdd');

if strcmp(fecha,data)==1
    j = msgbox('Los ficheros ya está actualizada','Actualización','warn');
    uiwait(j);
    disp('Computar');
else
    ex=fopen('actualizacion.dat','w');
```

```
fprintf(ex,'%s',fecha);
fclose(ex);

h = msgbox('La actualización tardará unos minutos en descargar y computar
predicción','Descarga y computación predicción','warn');
uiwait(h);
bp=waitbar(0,'ACTUALIZANDO...');
fecha=date;
fecha=datestr(fecha,'yyyymmdd');
carpeta=strcat('gfs.',fecha,'00');
cadena=strcat('/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f00');
mw = ftp('ftp.ncep.noaa.gov');
waitbar(1/12,bp,'Descargando ficheros');
mget(mw,cadena);
cadena=strcat('/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f06');
mget(mw,cadena);
waitbar(2/12,bp,'Descargando ficheros');
cadena=strcat('/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f12');
mget(mw,cadena);
waitbar(3/12,bp,'Descargando ficheros');
cadena=strcat('/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f18');
mget(mw,cadena);
waitbar(4/12,bp,'Descargando ficheros');
cadena=strcat('/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f24');
mget(mw,cadena);
waitbar(5/12,bp,'Descargando ficheros');
cadena=strcat('/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f30');
mget(mw,cadena);
waitbar(6/12,bp,'Descargando ficheros');
cadena=strcat('/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f36');
mget(mw,cadena);
waitbar(7/12,bp,'Descargando ficheros');
cadena=strcat('/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f42');
mget(mw,cadena);
waitbar(8/12,bp,'Descargando ficheros');
cadena=strcat('/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f48');
mget(mw,cadena);
waitbar(9/12,bp,'Descargando ficheros');
cadena=strcat('/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f54');
mget(mw,cadena);
waitbar(10/12,bp,'Descargando ficheros');
cadena=strcat('/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f60');
```



```

mget(mw,cadena);
waitbar(11/12,bp,'Descargando ficheros');
cadena=strcat('/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f66');
mget(mw,cadena);
waitbar(12/12,bp,'Descargando ficheros');
close(mw);
close(bp);

end

bp=waitbar(0,'Realizando predicción...');
waitbar(1/2,bp,'Computando predicción');
computacion();

global tmaxhoy;
global tmaxman;
global tmaxpas;
global tminhoy;
global tminman;
global tminpas;
global pphoy;
global ppman;
global pppas;
global vvhoy;
global vvman;
global vvpas;
global dvhoy;
global dvman;
global dvpas;
waitbar(2/2,bp,'Finalizado');
close(bp);
set(handles.maxhoy,'String',num2str(tmaxhoy));
set(handles.minhoy,'String',num2str(tminhoy));
set(handles.maxman,'String',num2str(tmaxman));
set(handles.minman,'String',num2str(tminman));
set(handles.maxpas,'String',num2str(tmaxpas));
set(handles.minpas,'String',num2str(tminpas));
disp('fin');

```

FIG 5.10 SCRIPT DE DESACARGA DE MODELO GFS

Lo primero que se realiza es la descarga de los ficheros necesarios para crear los patrones de predicción, estos se obtiene via ftp desde el servidor de la NOAA [27], se trata

de las salidas del modelo GFS de las 00, 06, 12, 18 para realizar la predicción del primer día, las de 24, 30, 36 y 42 para el segundo día y 48, 54, 60 y 66 para el tercer día. Se trata de ficheros en formato GRIB2 por lo que la herramienta que se ha utilizado no puede leerlos, por ello hay que hacer uso de la *toolbox njtboox* [23] que permite leer este tipo de archivos. La herramienta está basada en java y se puede utilizar añadiéndola al *path* de MATLAB

Aunque para realizar la computación y creación de los patrones hay que recurrir a un script más manual que se muestra en la Figura 5.11:

```
function computacion()
%%Lectura del fichero
%%PARA el dominio Iberia de 1.5º de resolución
% lon=-10:1.5:5
% lat=44:-1.5:35
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load 'dymy';
% load psy';

fecha=date;
fecha=datestr(fecha,'yyyymmdd');
carpeta=strcat('gfs.',fecha,'00');
cadena=strcat('C:/Users/Miguel/Documents/MATLAB/MLToolboxcomp/MLToolbox_r2009091
7/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f00');

ncRef = cadena;

%%VARIABLE NIVEL 1000 HPA

ncObj = mDataset(ncRef);
var = 'Temperature';
varObj = getVar(ncObj, var);
%varData = varObj(1,1,113:3:131,341:3:371);
varData = varObj(1,1,94:3:112,341:3:371);
t001000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];
% t001000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

var = 'Relative_humidity';
```

```

varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
h001000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

var = 'Geopotential_height';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
z001000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];
%%Pasamos de altura geopotencial a geopotencial
z001000=(z001000*9.8);

var = 'U-component_of_wind';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
u001000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

var = 'V-component_of_wind';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
v001000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

%%LEEMOS EL DE SALIDA 06h
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

cadena=strcat('C:/Users/Miguel/Documents/MATLAB/MLToolboxcomp/MLToolbox_r2009091
7/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f06');
ncRef = cadena;
%%VARIABLE NIVEL 1000 HPA

ncObj = mDataset(ncRef);
var = 'Temperature';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
t061000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

```

```
var = 'Relative_humidity';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
h061000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

var = 'Geopotential_height';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
z061000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];
z061000=(z061000*9.8);

var = 'U-component_of_wind';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
u061000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

var = 'V-component_of_wind';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
v061000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

%%%SALIDA DE LAS 12

cadena=strcat('C:/Users/Miguel/Documents/MATLAB/MLToolboxcomp/MLToolbox_r2009091
7/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f12');
ncRef = cadena;
%%%VARIABLE NIVEL 1000 HPA

ncObj = mDataset(ncRef);
var = 'Temperature';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
t121000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];
```

```

var = 'Relative_humidity';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
h121000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

var = 'Geopotential_height';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
z121000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];
z121000=(z121000*9.8);

var = 'U-component_of_wind';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
u121000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

var = 'V-component_of_wind';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
v121000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

%%%SALIDA DE LAS 18

cadena=strcat('C:/Users/Miguel/Documents/MATLAB/MLToolboxcomp/MLToolbox_r2009091
7/pub/data/nccf/com/gfs/prod/',carpeta,'/gfs.t00z.pgrb2f18');
ncRef = cadena';
% ncRef = 'abril2011.grib';
%%%VARIABLE NIVEL 1000 HPA

ncObj = mDataset(ncRef);
var = 'Temperature';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
t181000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

var = 'Relative_humidity';

```

```
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
h181000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

var = 'Geopotential_height';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
z181000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];
z181000=(z181000*9.8);

var = 'U-component_of_wind';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
u181000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

var = 'V-component_of_wind';
varObj = getVar(ncObj, var);
varData = varObj(1,1,94:3:112,341:3:371);
v181000=[varData(1,1:11) varData(2,1:11) varData(3,1:11) varData(3,1:11) varData(5,1:11)
varData(6,1:11) varData(7,1:11)];

pred=[z001000 z061000 z121000 z181000 t001000 t061000 t121000 t181000 u001000
u061000 u121000 u181000 v001000 v061000 v121000 v181000 h001000 h061000 h121000
h181000];
%Patron de comparación
% lon=-10:1.5:5
% lat=44:-1.5:35
pred=double(pred);
pat=carga_patron();

prediccions=[pred;pat];
dmn=readDomain('gfs');
[EOF,PC,MN,DV,PEV]=computeEOF(prediccions,'npc',50,'path',dmn.path);
```

```
[EOF,CP]=getEOF(dmn,'ncp',50);

%%%Ya podemos simular resultados en las redes neuronales
global tmaxhoy;
global tmaxman;
global tmaxpas;
global tminhoy;
global tminman;
global tminpas;
global pphoy;
global ppman;
global pppas;
global vvhoy;
global vvman;
global vvpas;
global dvhoy;
global dvman;
global dvpas;
%%%Temperatura maxim
load 'nntmax';

X=CP(:,:);
[X,mX,dX]=prestd(X);
P=sim(nnet,X(:,1));
Pred=((P.*repmat(dY,[1 size(P,2)]))+repmat(mY,[1 size(P,2)]));
tmaxhoy=Pred;
```

FIG 5.11 SCRIPT COMPUTACIÓN Y PREDICCIÓN

Este *script* lee los archivos descargados y lee las variables en las coordenadas del patrón, generando un vector similar a los del entrenamiento de la red neuronal, después de eso se carga la red neuronal, se estandariza el vector patrón y se simula para que nos dé el resultado de la predicción. En concreto con el código presentado anteriormente se estaría realizando la predicción de la temperatura máxima para el primer día para el patrón con componentes temporales “iberiatemp” que es relativamente sencillo y da buenos resultados.

Una vez se han calculado los datos de predicción para cada día se muestran en la siguiente pantalla (Figura 5.12):

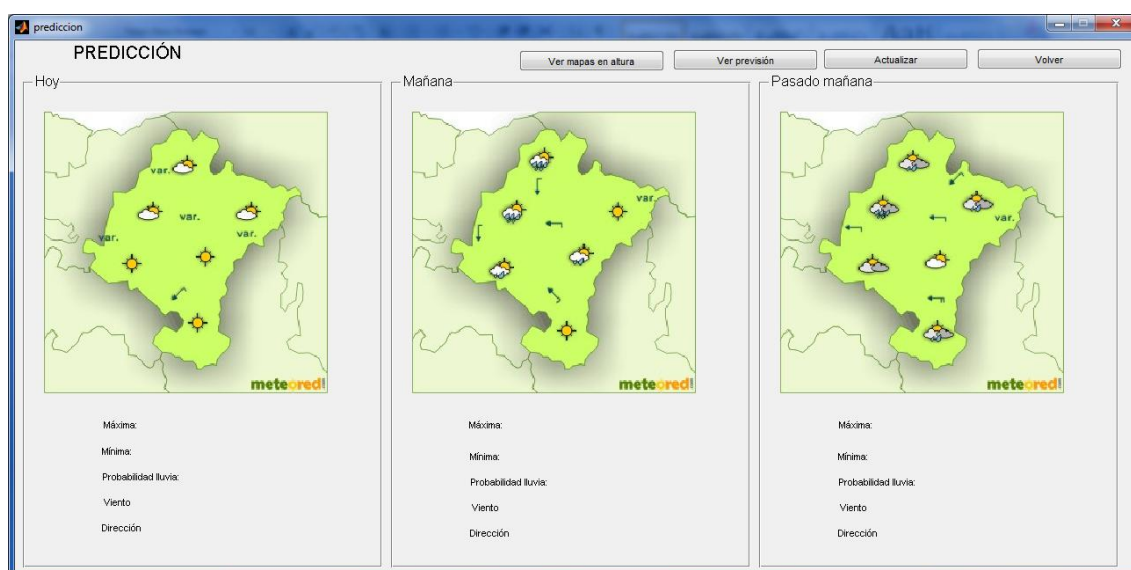


FIG 5.12 PREDICCIÓN

También se puede acceder desde esta pantalla (Figura 5.13) a una visualización de los mapas en altura previstos para las variables:

The screenshot shows a software interface titled 'Previsión altura'. It features a section titled 'Variables a visualizar'. This section is divided into two main areas. The first area, labeled 'Altura', contains five radio buttons for different pressure levels: 1000 hPa (selected), 950 hPa, 850 hPa, 500hPa, and 300 hPa. The second area, labeled 'Variable', contains five radio buttons for different meteorological variables: Geopotencial, Temperatura, Humedad relativa, Velocidad u viento (selected), and Velocidad v viento. To the right of these sections is a section labeled 'Horas' with the sub-label 'Horas a partir de las 00'. It contains twelve radio buttons for different times of day: 00, 06, 12, 18, 24, 30, 36, 42, 48, 54, 60, and 66 (selected). At the bottom of the window, there are two buttons: 'Continuar' and 'Volver'.

FIG 5.13 SELECCIÓN VARAIBALES PREVISIÓN MAPAS ALTURA

También se puede actualizar la previsión, se descargarán los nuevos ficheros y se computará (sólo se descargarán nuevos ficheros en el caso de que no estén actualizados a

día de hoy). Este es un proceso que puede llevar un buen rato por lo que se advierte antes de iniciarlo:

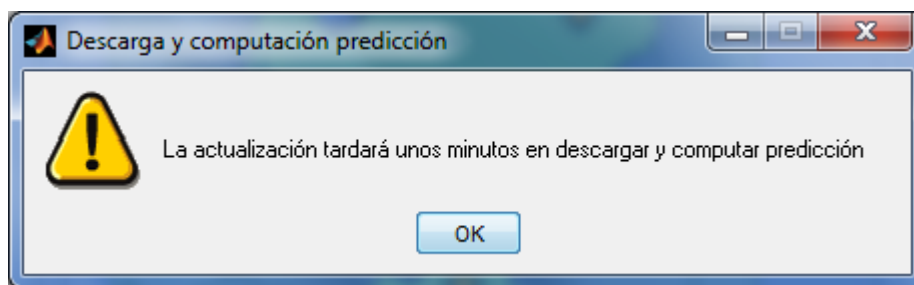


FIG 5.14 CUADRO DE DIÁLOGO ADVERTENCIA

Y se va actualizando una *waitbar* como se ve en la Figura 5.15:

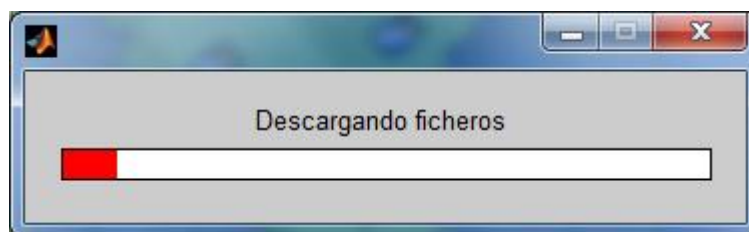


FIG 5.15 WAITBAR DESCARGA Y COMPUTACIÓN

6-CONCLUSIONES

En este proyecto se ha desarrollado el estudio experimental de patrones meteorológicos comparando y viendo los resultados para las redes neuronales y para regresión. También se ha integrado todo en una aplicación con interfaz gráfico para un manejo más fácil de todos los datos utilizados, cumpliendo los objetivos que se habían planteado al comienzo del mismo.

Primeramente se extrae como conclusión que el manejo de los patrones meteorológicos a la hora de computar los componentes principales es un proceso que requiere de una capacidad de cómputo y memoria elevados, ya que en casos con una gran cantidad de puntos de rejilla y varias variables el proceso con un PC normal (4GB de Memoria, intel Core i5 2.4 Ghz), puede tardar varias horas e incluso dejar al ordenador sin memoria a la hora de computar la matriz de covarianza. Por lo tanto para este tipo de trabajos se requieren computadores con gran capacidad de cómputo que realicen este trabajo de manera más rápida.

Por otro lado, hablando ya de las pruebas, se puede concluir que los resultados a pesar de no ser espectaculares, sí que son bastante buenos ya que la meteorología es un tema complicado y un pequeño cambio puede desencadenar cambios importantes.

- Para las temperaturas se puede hacer predicción determinista ya que el error medio es de 1'3 grados a así que es bastante bueno utilizando redes neuronales, sin embargo, es uno de los pocos casos que se ha visto mejorado por la regresión lineal múltiple con la que se obtiene un error medio de 1 grado.
- Para la lluvia el error es mayor y se predice de peor manera, por lo que se opta por hacer una predicción probabilística al estilo de la página de AEMET con la predicción de municipios (si se pone un umbral del 50% el acierto en las predicciones es de un 84%).
- Para el viento pasa parecido que con la lluvia, en este caso se clasifica en dos tipos de viento con un umbral de velocidad acertando sobre el 85% de las veces.
- Para la dirección también he hecho un clasificador en 4 clases de viento que tiene un acierto superior al 89%.

A la hora de valorar los diferentes parámetros de entrenamiento, funciones de salida ... sí que se ha encontrado diferencia en los métodos de entrenamiento ya que los dos que mejores resultados dan con diferencia sobre los demás son “traingdx” y ”levenberg-marquardt”, entre estos dos hay diferencias en el tiempo de computación y consumo

memoria, el primero necesita más iteraciones para llegar a un resultado parecido pero consume muchos menos recursos, por lo que para redes neuronales más complejas es mejor. Por otro lado el de "levenberg-marquardt", alcanza resultados mejores con menos iteraciones pero consume más memoria y en redes complejas es muy lento. Al ir avanzando con las pruebas se ha utilizado más el "traingdx".

Lo que realmente marca la diferencia entre unos clasificadores u otros son los patrones, hay que escoger bien las variables para poder realizar una predicción mejor, estos patrones nos deben aportar información sobre lo que se predice.

La comparación entre redes neuronales y regresión lineal es favorable a las redes neuronales, con la mayoría de patrones dan mejores resultados y lo hacen con muchos menos componentes principales, ya que en las pruebas de regresión se utilizan todos los componentes, si se utilizan menos funcionan de peor manera. Para las redes neuronales puede valer con 50 CPs.

El caso de la temperatura máxima sí que se obtienen muy buenos resultados mediante la regresión lineal múltiple y eso es debido a la relación lineal con los patrones.

Una vez se tiene los resultados resulta sencillo integrar todo lo utilizado y crear una herramienta que facilite el uso de todo lo aplicado durante el proyecto mediante un interfaz gráfico al que la gente está más acostumbrada.

7-ANEXO I

En este pequeño apartado se va mostrar el material que se ha desarrollado en el proyecto y como habría que utilizarlo.

7.1- MATERIAL ADJUNTO

Junto con la memoria se adjunta todo el material utilizado y desarrollado durante la realización del proyecto.

Todo el material se encuentra dentro de la carpeta “MLTtoolboxcomp”, dentro de ella se encuentran las toolbox utilizadas, los scripts de las pruebas y la herramienta con interfaz gráfico.

7.2- EJECUCIÓN

Antes de realizar la ejecución hay que asegurarse de tener en el path de MATLAB varios archivos, por lo que hay que cambiar los *scripts*.

En este caso hay que realizarlo en los *scripts*:

- Rutanj: con la ruta de los archivos de java necesarios para la toolbox njtoolbox.
- Predicción: cambiar la ruta donde se guardan los archivos del GFS.
- Computación: cambiar la ruta a la que se ha puesto para guardar los archivos de GFS.
- Init: poner la ruta donde se guardarán las imágenes.

Una vez se tienen correctos todos los paths solamente hay que entrar en la carpeta meteolab y ejecutar el script init.

Para ejecutar la herramienta gráfica hay que volver hacia a la carpeta anterior y ejecutar el script meteo.

8-ANEXO II

Utilizando la toolbox de matlab se pueden crear, entrenar y simular las redes neuronales necesarias para realizar los experimentos. Estos son los parámetros básicos utilizados:

8.1-DECLARACIÓN DE LA TOPOLOGÍA DE LA RED

Para declarar la arquitectura de la red e inicializar de forma aleatoria sus pesos y umbrales utilizaremos la función NEWFF:

newff (new feed-forward network): Crea una red con conexiones hacia delante y con retropropagación de errores.

Sintaxis:

```
net=newff(PR,[S1 S2 ...SN1],{TF1 TF2 ...TFN1},BTF)
```

Descripción:

net - nombre asignado a la red

PR - es una matriz (Rx2) donde aparecen los máximos y mínimos de los R elementos de entrada.

Si – tamaño de la capa iésima, para un total de N1 capas

TFi – Función de transferencia de la capa iésima. Por defecto = tansig (otras posibilidades: logsig o purelin).

BTF – Algoritmo de retropropagación de error. Por defecto = traingdx (otras posibilidades: traingd, traingdm, traingda).

8.2-ENTRENAMIENTO

Para entrenar una red en Matlab se utiliza la función *train*. Dicha función utiliza como argumentos el nombre de la red que se quiere entrenar (y que anteriormente ha debido ser creada) y los patrones, compuestos por unos vectores de entradas (unidos en una matriz “a”) y sus correspondientes salidas (todas ellas en una matriz “p”). De esta manera la línea de código que se debería escribir es:

```
[net,pr] = train (net,a,p);
```

Se pueden utilizar diferentes tipos de entrenamiento para las redes del tipo Perceptrón multicapa:

- Traingd: regla delta generalizada.
- Traingdm: regla delta generalizada con *momentum*.
- Traingdx: Ratio de aprendizaje variable con *momentum*.
- Levenberg-Marquardt

8.3-SIMULACIÓN DE LA RED

Para obtener las salidas de una determinada red ante unas ciertas entradas se utiliza la función `sim`. Esta función devuelve un vector cuyas componentes son las salidas que se obtienen de cada neurona de salida de la red. A dicha función se le introducen como argumentos el nombre del objeto que representa a la red en Matlab, y el vector de entradas a la red.

Se pueden obtener varias simulaciones de un tiempo introduciendo una matriz, cuyos vectores sea cada uno de los vectores de entrada de los que se quiere obtener su salida. De esta manera, si se tiene una red “net” de la que se quiere obtener su respuesta ante unas entradas “a”, se escribe en Matlab la siguiente línea de código.

$$p = \text{sim}(\text{net}, a);$$

Para obtener más información acerca del toolbox de redes neuronales de Matlab se puede consultar su guía de usuario [10].

9-REFERENCIAS Y BIBLIOGRAFÍA

- [1] GRIded Binary, <http://www.grib.us/> [2011]
- [2] MATLAB, <http://www.mathworks.es/products/matlab/>
- [3] J.M. Gutiérrez, R. Ancell, A.S. Cofiño, C.M. Sordo, *Redes probabilísticas y neuronales en las ciencias atmosféricas*, Ministerio Español de Medio Ambiente, 2004.
- [4] AEMET. Agencia Estatal de Meteorología. <http://www.aemet.es/>
- [5] METEORED, <http://www.tiempo.com/>
- [6] ELTIEMPO.ES, <http://www.eltiempo.es/>
- [7] METEOSIM, http://www.meteosim.com/es/prediccion_por_ciudades/
- [8] MASS, <http://www.ceam.es/ceamet/modelos/MASS/mass.html>
- [9] Manolis I. A. Lourakis ,“A Brief Description of the Levenberg-Marquardt Algorithm Implemented”
- [10] MATEMATICAS M&M, DISCRIMINANTE O HESSIANO Para encontrar máximos, mínimos y puntos de silla en funciones de varias variables.
- [11] 050 METEOROLOGY, Jeppesen 2001.
- [12] Neural Network Toolbox 7User’s Guide, http://www.mathworks.com/help/pdf_doc/nnet/nnet.pdf
- [13] James A. Freeman, David M. Skapura, *Neural Networks Algorithms, Applications, and Programming Techniques*, Addison Wesley, 1991.
- [14] Atiya, A.F.; El-Shoura, S.M.; Shaheen, S.I.; El-Sherif, M.S.; , "A comparison between neural-network forecasting techniques-case study: river flow forecasting," *Neural Networks, IEEE Transactions on* , vol.10, no.2, pp.402-409, Mar 1999
- [15] Paras, Sanjay Mathur, Avinash Kumar, and Mahesh Chandra, “A Feature Based Neural Network Model for Weather Forecasting, World Academy of Science, Engineering and Technology, 34, 2007.
- [16] Rafael Ancell Trueba, *Aportaciones de las Redes Bayesianas en Meteorología. Predicción Probabilística de Precipitación*, 2009.

- [17] T. Hall, H.E. Brooks, C. A. Doswell, *Precipitation Forecasting Using a Neural Network*, 1997.
- [18] Datos estación de pamplona y fenómenos meteorológicos, <http://meteo.navarra.es>
- [19] Datos estación Pamplona
http://www.tutiempo.net/clima/Pamplona_Noain/1976/80850.htm
- [20] Smith, Scott T. (2006). Matlab advanced gui development .Dog Ear Publishing.
- [21] Javier García Galón. (2007) *Aprenda MatLab 7.0 como si estuvieras en primero*. Universidad Politécnica Madrid Ediciones
- [22] International verification methods workshop
http://www.cawcr.gov.au/projects/verification/#Methods_for_spatial_forecast
- [23] Netcdf-Java Toolbox
<http://njtbx.sourceforge.net/njdocs/njtbxhelp/njtbxhelp.html>
- [24] Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. Monthly Weather Review, 78:1–3.
- [25] European Centre for Medium-Range Weather Forecasts
<http://www.ecmwf.int/>
- [26] Meteolab. Meteorological machine learning toolbox for matlab
<http://grupos.unican.es/ai/meteo/MeteoLab.html>
- [27] Salidas GFS Nco-Noaa
<ftp://ftp.ncep.noaa.gov/pub/data/nccf/com/gfs/prod/>
<http://www.nco.ncep.noaa.gov/pmb/products/gfs/>



**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN**

INGENIERÍA INFORMÁTICA

**PFC -“RECONOCIMIENTO DE PATRONES
METEOROLÓGICOS MEDIANTE TÉCNICAS
ESTADÍSTICAS Y NEURONALES”**

Miguel Irigoyen Mancho

Índice

1-Objetivos

2-Metodología utilizada

3-Otras formas de abordar el problema

4-Descripción pruebas

- Variables
- Patrones
- Predicciones
- Errores...

5-Resultados experimentales

6-Integración en herramienta

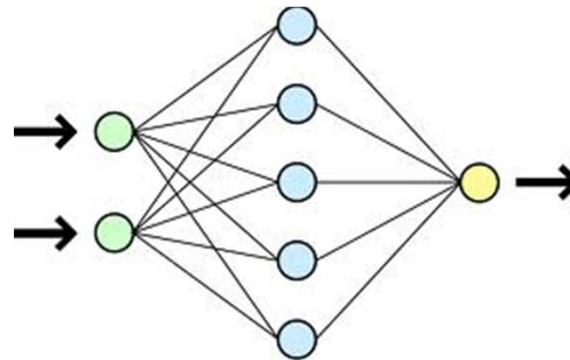
7-Conclusiones

1- Objetivos

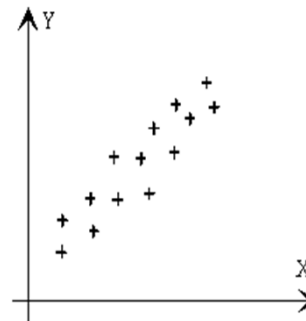
- Obtener y definir los patrones que caractericen el estado de la atmósfera.
- Obtener los datos de la estación meteorológica con los que se emparejan los patrones.
- Realizar pruebas con redes neuronales variando parámetros.
 - Crear scripts
 - Valoración de error cometido en distintos tipos de predicción
 - Gráficas...
- Integración en herramienta con interfaz gráfico.
- Tanto las pruebas como la herramienta se realizarán en MATLAB.

2- Metodología utilizada

- Se emplean redes neuronales de distinto tipo para el reconocimiento de patrones.



- También se realizan pruebas con regresión.



3- Otras formas de abordar el problema

- Otros métodos utilizados para la predicción local
 - La correlación canónica.
 - Método de análogos.
 - Redes probabilísticas
 - Redes Bayesianas.
 - Redes de Markov.

3- Herramientas existentes

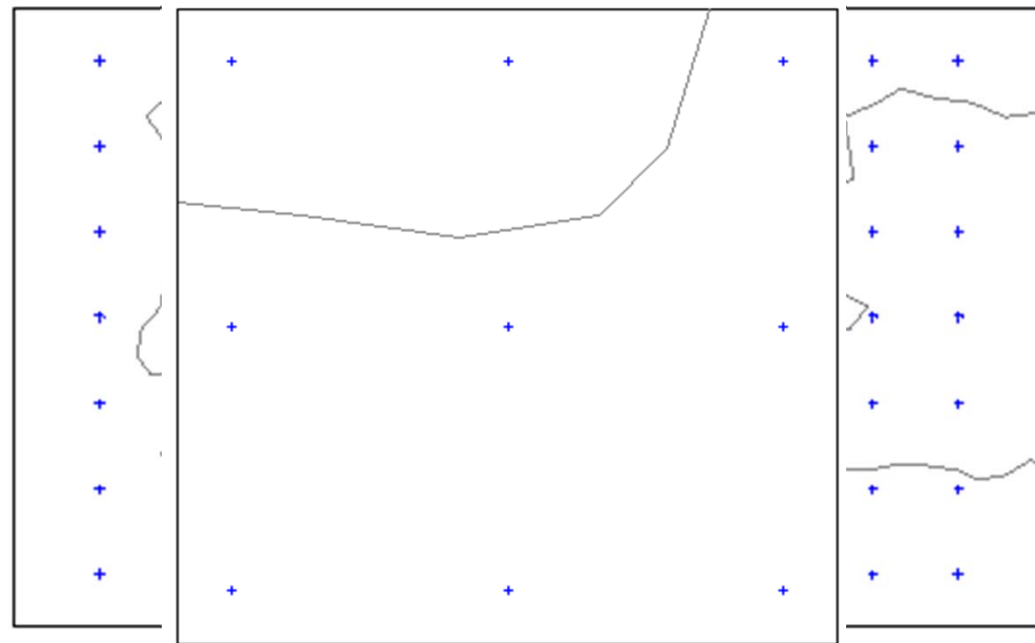


4- Descripción pruebas

- Se predicen las siguientes variables:
 - Temperatura máxima
 - Temperatura mínima
 - Pprecipitación
 - Velocidad viento
 - Dirección viento
- Para ello es necesario obtener valores históricos de Pamplona, se obtienen de la web Meteorología y climatología de Navarra (Gobierno de Navarra).

4- Descripción pruebas

- Se obtienen y crean los patrones que caracterizan el estado de la atmósfera.
- Archivos en formato GRIB (ECMWF).



NH@ria

4- Descripción pruebas

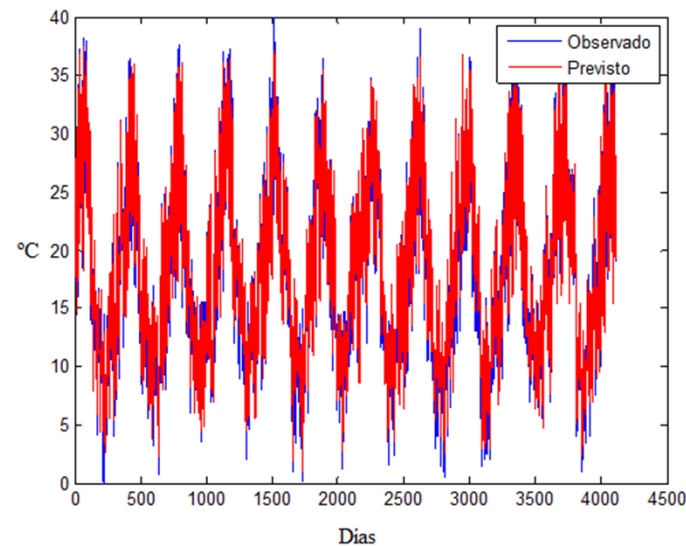
- Para cada punto de la rejilla se tienen los valores de las variables:
 - Geopotencial (Z)
 - Temperatura(T)
 - Velocidad viento U: componente zonal W-E
 - Velocidad viento V: componente meridional N-S
 - Humedad relativa (H)
- A su vez se dispone de los valores a diferentes niveles de presión:
 - 1000 hPa
 - 850 hPa
 - 500 hPa ...

4- Descripción pruebas

- Tras realizar una serie de pruebas se decide realizar los siguientes tipos de pruebas:
 - Predicción determinista para las temperaturas (Ej: $T_{\text{max}} = 24^{\circ}$).
 - Predicción probabilística para la precipitación
 - ✦ Predicción del evento
 - ✦ En porcentaje
 - Predicción probabilística para la velocidad del viento con umbral.
 - Clasificación en 4 clases para la dirección del viento.

4- Descripción pruebas

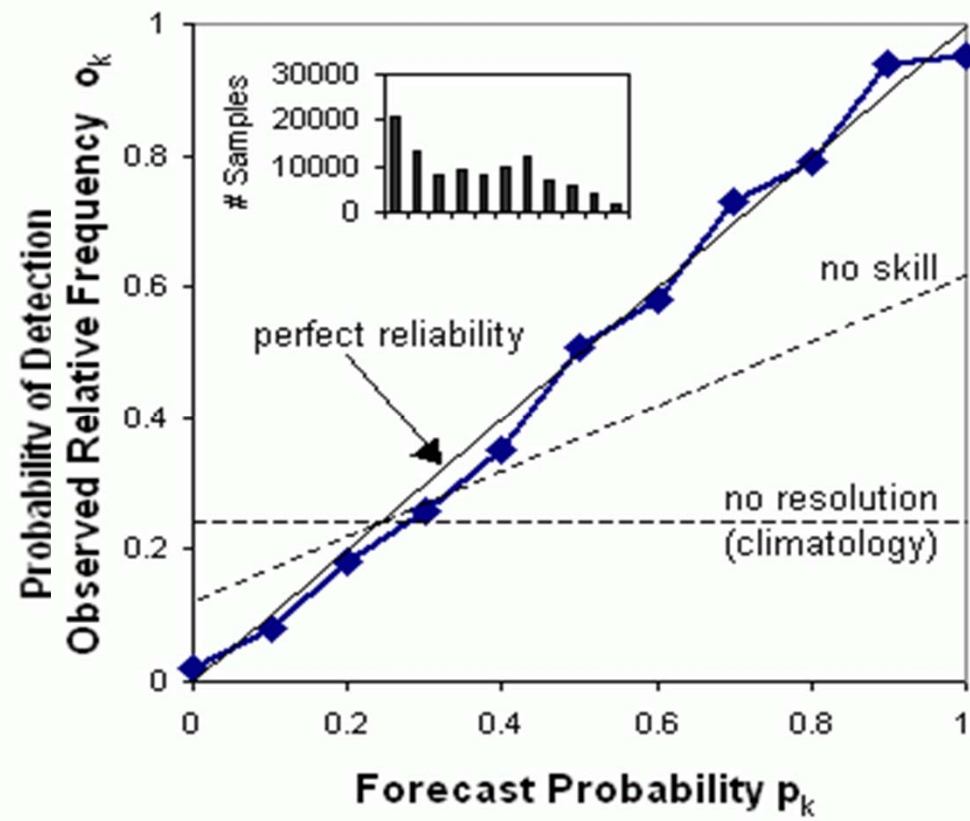
- Se valora el error de diferentes formas dependiendo del tipo de predicción
- Para la previsión determinista:
 - Error cuadrático medio en entrenamiento
 - Error cuadrático medio de test
 - Error absoluto medio
 - Gráficos:



4- Descripción pruebas

- Para la predicción probabilística se utilizan diferentes índices:
 - Brier score: error medio calculado en unidades de probabilidad para la ocurrencia de las distintas categorías de la variable (0 es óptimo).
 - Brier skill score: Mide la mejora en sistemas probabilísticos con respecto a un sistema de referencia de predicción (1 es perfecto).
 - RSA: El área ROC varía entre 0 y 1, siendo 1 el mejor resultado y 0.5 el peor.
 - Comparación directa con umbral al 50 %.
 - Gráficos (siguiente).

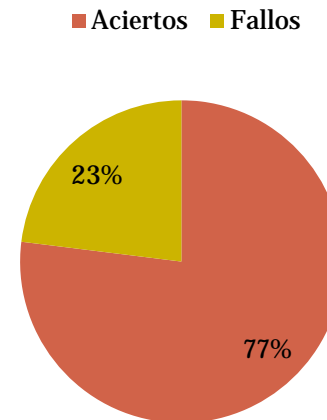
4- Descripción pruebas



4- Descripción pruebas

- Para la clasificación en clases se pone como salida un vector con el índice de la clase a 1 y el resto a 0
- Se coge el máximo y se compara con lo observado
 - Porcentaje de acierto

Porcentaje aciertos

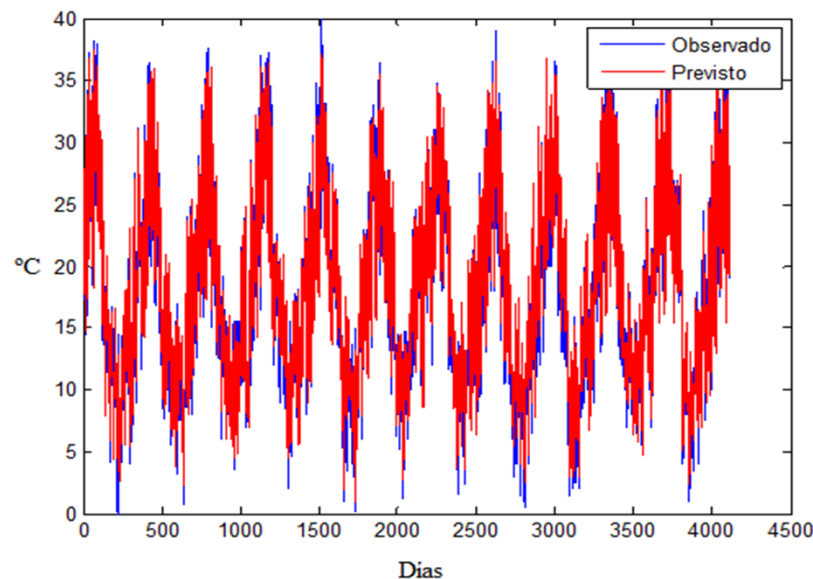


5-Resultados experimentales

- Se realizan pruebas variando diferentes parámetros de la red neuronal
 - Número de capas
 - Número de neuronas en cada capa
 - Se varía el algoritmo de entrenamiento : traingdx, levenberg-Marquardt...
 - Funciones de salida
- Se prueba con diferentes patrones
 - Seleccionan diferentes variables
 - Diferentes niveles de presión
 - Escala temporal
- Regresión lineal múltiple con algunos patrones

5-Resultados experimentales

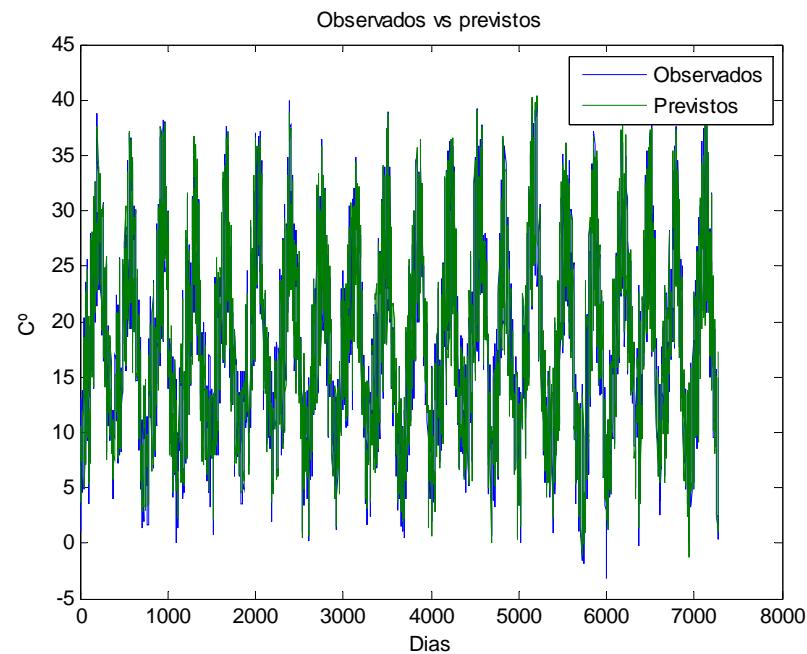
- Para la temperatura máxima con una red de **dos capas ocultas con 8 y 5 neuronas**, con entrenamiento **traingdx**. **Patrón Iberia 1º**



Error entrenamiento	Error test	Error medio
0.0463	0.0496	1.3649

5-Resultados experimentales

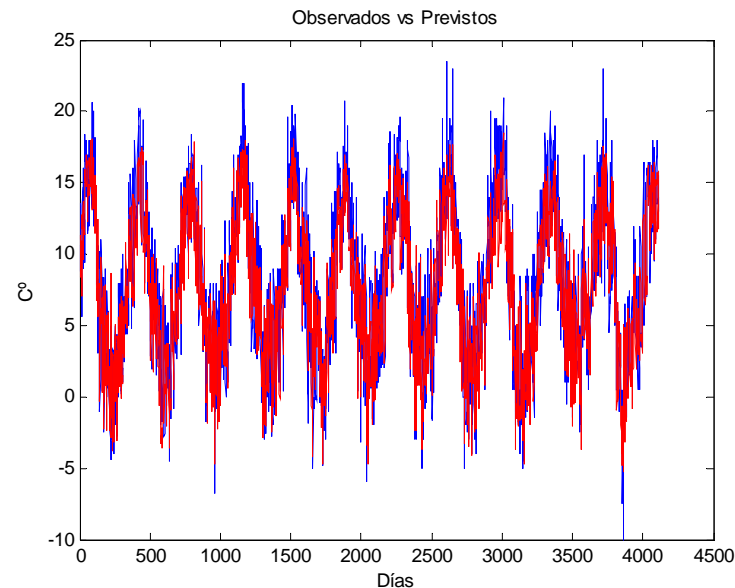
- Con el patrón iberiatemp es el único caso en el que se mejora con la regresión



Con un error medio de 1° !!

5-Resultados experimentales

Para la mínima con los mismo parámetros de red:

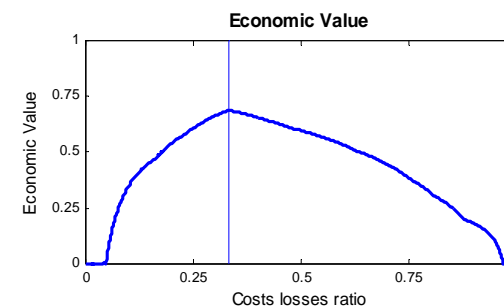
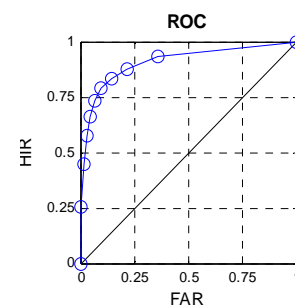
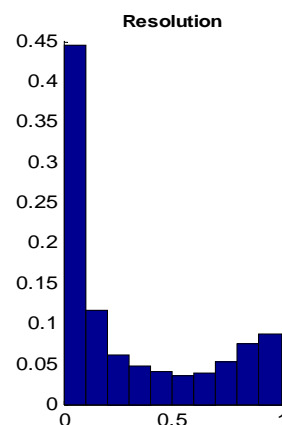
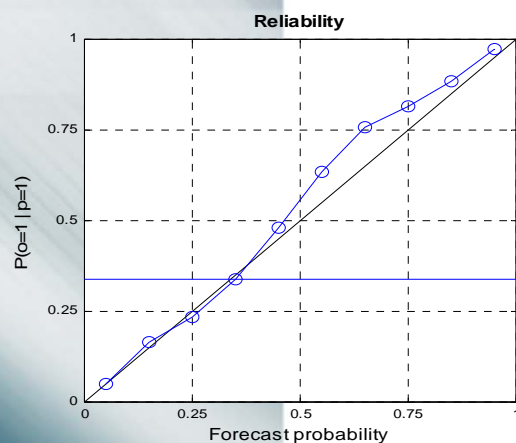


Error entrenamiento	Error test	Error medio
0.1039	0.1255	1.5590

5-Resultados experimentales

- Para la precipitación
- Patrón con componentes temporales sobre 1000hPa
- Red con dos capas ocultas y traingdx.

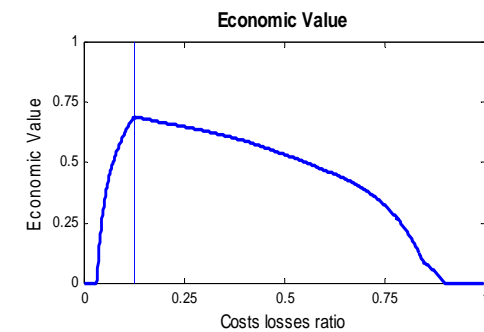
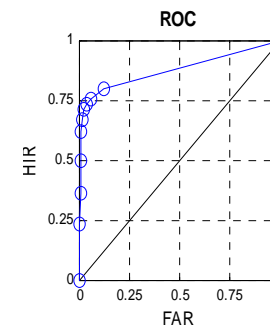
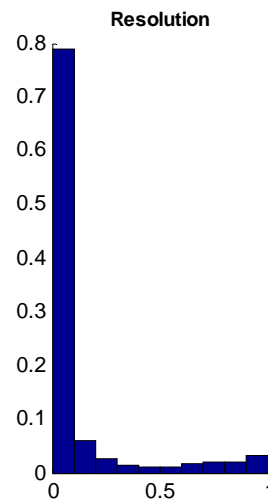
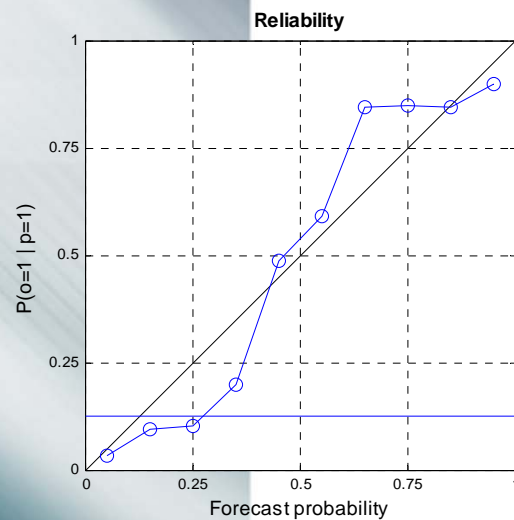
Error entrenamiento	BSS	Porcentaje
0.0942	0.5683	83.2325
BSC	BSP	RSA
0.2227	0.0961	0.8382



5-Resultados experimentales

- Para el viento con un patrón iberiatemp y una red de dos capas ocultas.

Error entrenamiento	BSS	Porcentaje
0.0362	0.5301	85.0109
BSC	BSP	RSA
0.1099	0.0516	0.7467

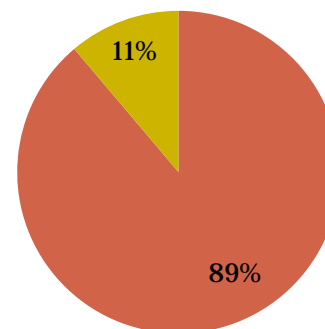


5-Resultados experimentales

- Para el patrón de IberiaUV con componentes temporales se obtiene un acierto del 89.8144% para la dirección.

Porcentaje aciertos

■ Aciertos ■ Fallos



6-Integración herramienta

- Se va a proceder a integrar todo lo utilizado en una herramienta con interfaz gráfico.



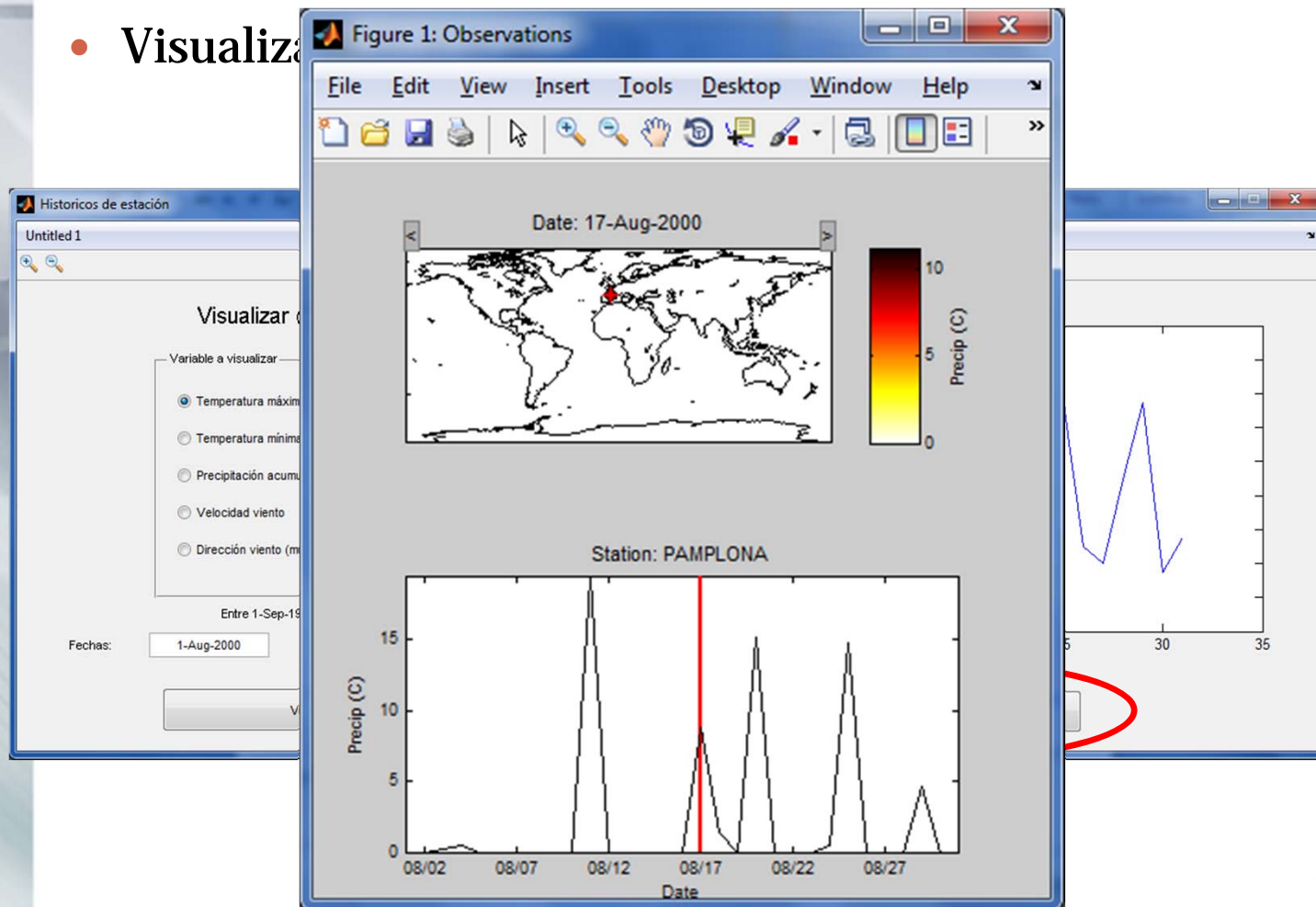
6-Integración herramienta



- Vi
 ○ Tabla
 ○ Gráfico
- Localización estación.
- Mapa predicción.

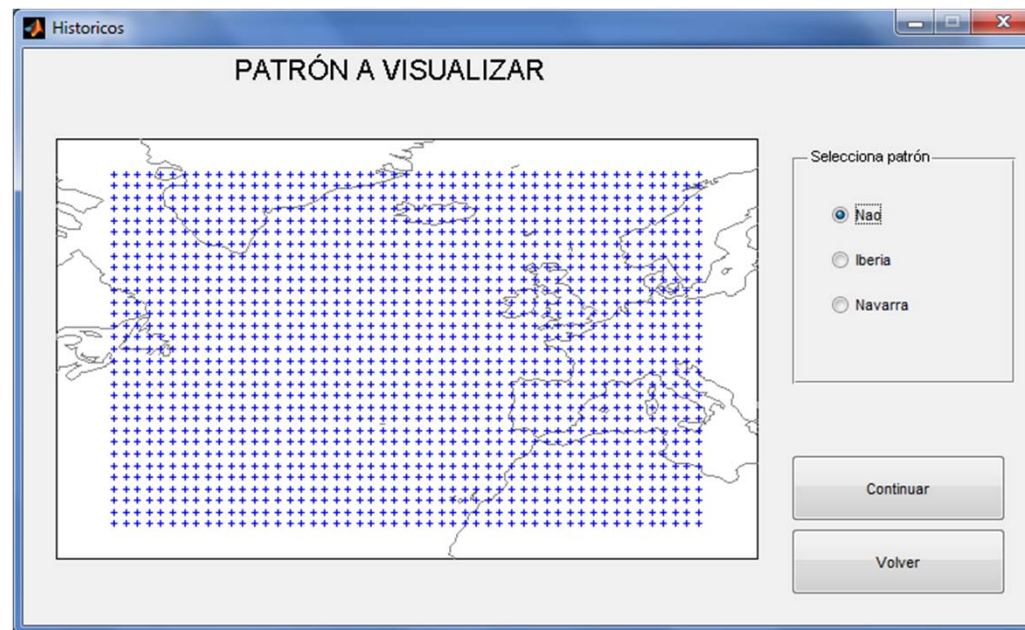
6- Integración herramienta

- Visualiza



6- Integración herramienta

- Visualización patrones



6- Integración herramienta

- Selección de variables a visualizar

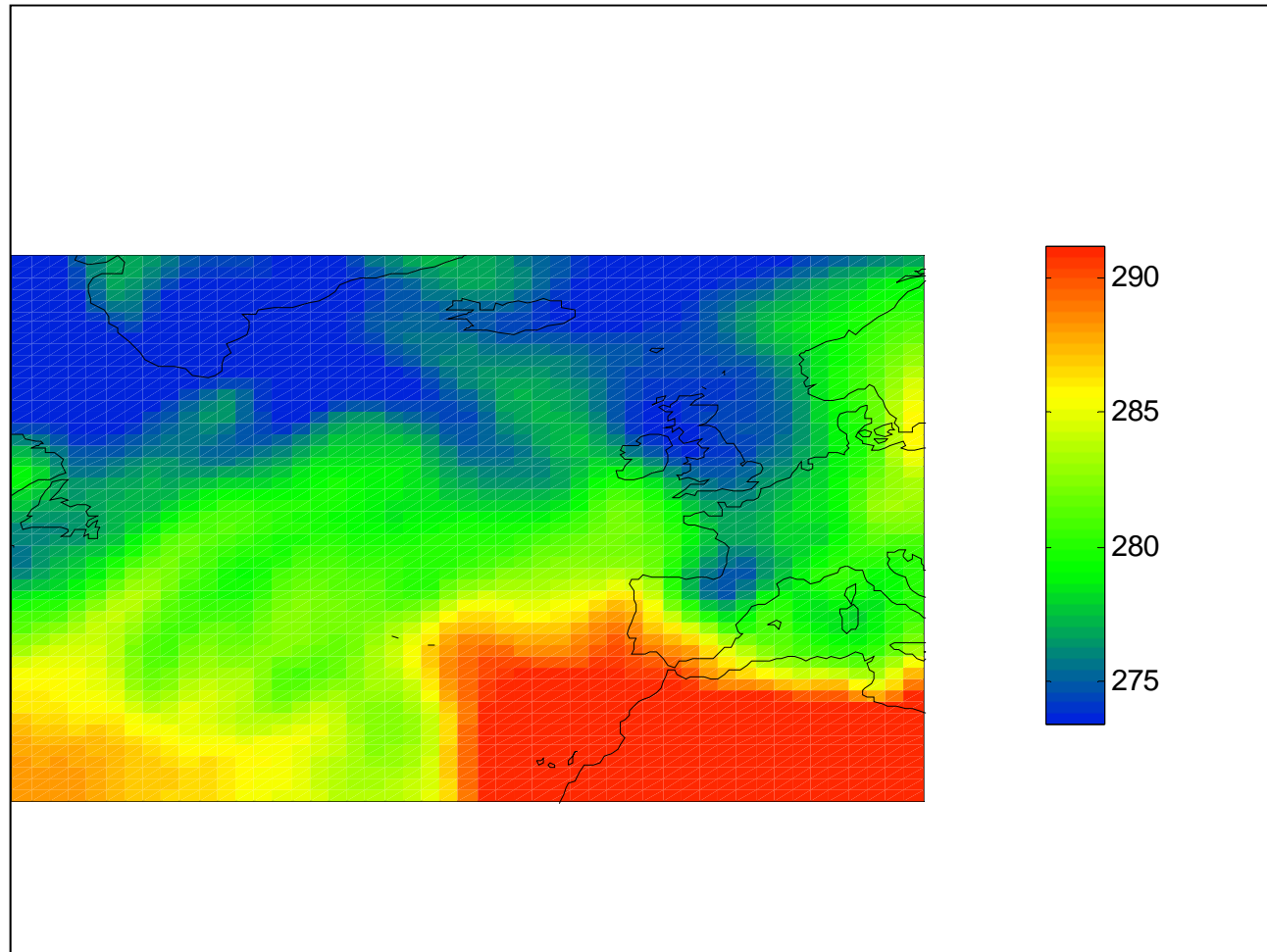
The screenshot shows a software window titled "naosel" with the subtitle "Variables a visualizar". Inside the window, there are two main sections for selection:

- Altura:** Two radio buttons are present. The first is labeled "850 hPa" and is selected (indicated by a blue dot). The second is labeled "500hPa" and is unselected.
- Variable:** A list of five radio buttons. The first four are "Geopotencial", "Temperatura", "Humedad relativa", and "Velocidad u viento". The fifth is "Velocidad v viento". The "Velocidad u viento" option is selected.

Below these sections, there is a date range specification: "Fechas: desde 1-Jan-1989 hasta 31-Dec-2009". Underneath this, there are two input fields labeled "Desde" and "hasta", both of which are currently empty.

At the bottom of the window, there are two buttons: "Continuar" on the left and "Volver" on the right.

6- Integración herramienta



6- Integración herramienta

- Predicción
- Se realiza la predicción a 3 días para las variables que se han pronosticado en las pruebas.
- Se emplean dos *scripts* en MATLAB que realizan todo el proceso.
- Se obtienen los modelos numéricos de predicción del modelo americano GFS.

6- Integración herramienta

Descarga de
ficheros GRIB
via FTP



Obtención de
variables



Crear patrones
similares a lo
de
entrenamiento



Computación
CPs



Simular en red
entrenada (o
regresión).



RESULTADOS

6- Integración herramienta



7-Conclusiones

- Se ha desarrollado el estudio experimental de patrones meteorológicos comparando y viendo los resultados para las redes neuronales y para regresión.
- Se ha integrado todo lo utilizado en una herramienta con GUI.
- En cuanto a los resultados:
 - Para las temperaturas se puede hacer predicción determinista ya que el error medio es de 1'3 grados a así que es bastante bueno utilizando redes neuronales.
 - Para la lluvia el error es mayor y se predice de peor manera, por lo que se opta por hacer una predicción probabilística.

7-Conclusiones

- Para el viento pasa parecido que con la lluvia, en este caso se clasifica en dos tipos de viento con un umbral de velocidad acertando sobre el 85% de las veces.
- Para la dirección también he hecho un clasificador en 4 clases de viento que tiene un acierto superior al 89%.
- Son resultados bastante buenos ya que la meteorología es un tema complicado y un pequeño cambio puede desencadenar cambios importantes.
- Diferencias de resultados originadas por patrones.
- Traingdx y Levenberg-Marquardt mejores métodos de entrenamiento.
- Necesidad de gran capacidad de cómputo.
- Integración en herramienta para facilitar uso.